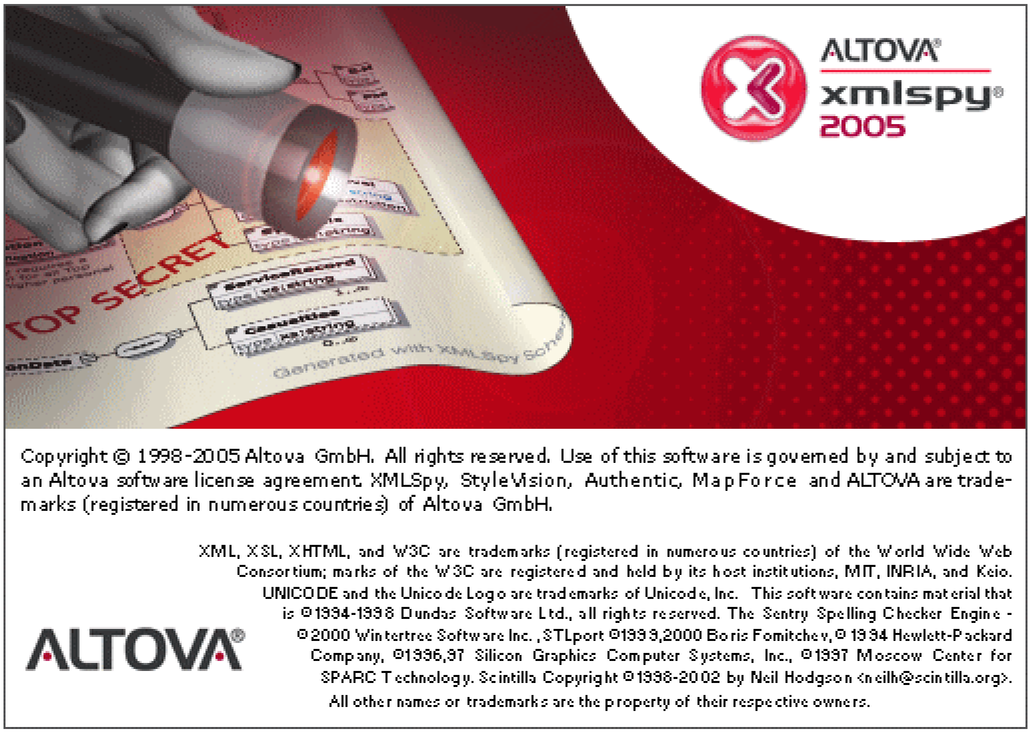
User Manual


ALTOVA
xmlspy
2005
TOP SECRET
ServiceRecord
Casualties
Generated with XMLSpy Schema
ALTOVA

XMLSpy 2005 Home Edition

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Published: 2005

Printed in the United States of America

5 4 3 2 1

ISBN 1-933210-00-1

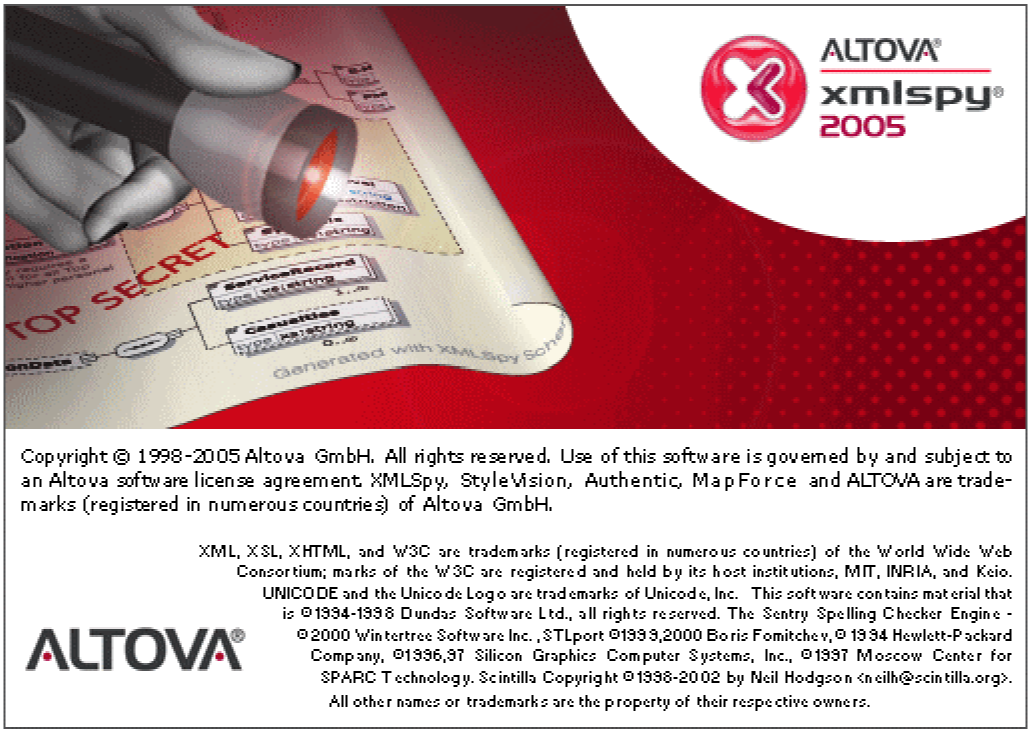
Table of Contents

XMLSpy 2005 Home Edition

Welcome to XMLSpy 2005 Home Edition

1 Welcome to XMLSpy 2005 Home Edition

Altova® XMLSpy® 2005 Home Edition is the industry standard XML Development Environment for designing, editing and debugging enterprise-class applications involving XML, XML Schema, XSL/XSLT, SOAP, WSDL and Web service technologies. It is the ultimate productivity enhancer for J2EE, .NET and database developers.



Copyright © 1998-2005 Altova GmbH. All rights reserved. Use of this software is governed by and subject to an Altova software license agreement. XMLSpy, StyleVision, Authentic, MapForce and ALTOVA are trademarks (registered in numerous countries) of Altova GmbH.

XML, XSL, XHTML, and W3C are trademarks (registered in numerous countries) of the World Wide Web Consortium; marks of the W3C are registered and held by its host institutions, MIT, INRIA, and Keio. UNICODE and the Unicode Logo are trademarks of Unicode, Inc. This software contains material that is ©1994-1998 Dundas Software Ltd., all rights reserved. The Sentry Spelling Checker Engine - ©2000 Wintertree Software Inc. ,STLport ©1999,2000 Boris Fomitchev, © 1994 Hewlett-Packard Company, ©1996,97 Silicon Graphics Computer Systems, Inc., ©1997 Moscow Center for SPARC Technology. Scintilla Copyright ©1998-2002 by Neil Hodgson <neilh@scintilla.org>. All other names or trademarks are the property of their respective owners.

Introduction

2 Introduction

This introduction briefly describes:

- Altova's various XML products and how they relate to each other
- XMLSpy's main features
- The user interface

This section is intended to serve as a general introduction to XMLSpy 2005, and will familiarize you with the product's capabilities and interface.

2.1 Altova's XML Products

Altova's XML products are easy to use and cover all your XML needs. They complement each other and provide you with a comprehensive XML application development environment. The Altova family currently comprises the following products.

Altova XMLSpy® 2005 is a comprehensive IDE for developing XML projects, and is available in three feature configurations. At the top end, the Enterprise Edition provides an efficient and flexible environment for creating and editing DTDs, XML Schemas, XML files, and XSLT stylesheets. It has powerful editing features, multiple document views—including Altova's own Authentic View—validation, and XSLT transformations with an internal processor. It can import, and export to, text files and databases. Among other features are management of XML documents in projects, an XSLT and XQuery Debugger, a WSDL Editor, and code generation. The Professional and Home Editions have different feature configurations.

Altova StyleVision® 2005 is a new approach to writing complex XSLT stylesheets using an intuitive drag-and-drop user interface. With StyleVision you also create Authentic Stylesheets that are used to control the display and data entry of XML documents in Authentic View. StyleVision 2005 is available in Enterprise and Professional editions.

Altova Authentic® 2005 (Desktop and Browser editions) are word-processor type editor programs which support form-based data input of XML documents. You can insert components such as graphics and tables, and validate in real-time against a schema.

Altova MapForce™ 2005 is a product for mapping one schema to another and converting XML files based on one schema to XML files based on another.

Altova website
You may also want to periodically check the Altova website, www.altova.com, for news, updates, additional documentation, and support.

Please note that user manuals for all Altova products are available in the following formats:

- Online manuals, accessed via the Support page at the Altova website
- Printable PDFs, which you can download from the Altova website and print locally
- Printed books that you can buy via a link at the Altova website

The documentation on the website is updated periodically and kept up-to-date with the current versions.

Support and feedback
If you require additional information or have a query about Altova products, please don't hesitate to visit our Support Center at the Altova website. Here you will find:

- Links to our FAQ pages
- Discussion forums on Altova products and general XML subjects
- Online Support Forms that will be processed by our support team

Also, please feel free to send us your feedback about the documentation.

2.2 XMLSpy's main features

XMLSpy 2005 is an integrated Development Environment (IDE) for the development of XML projects. XMLSpy 2005 can be used, among other things, to edit and process a variety of XML and other text documents and process documents with the built-in XSLT 1.0 processor, XSLT 2.0 processor and XQuery 1.0 processor.

XMLSpy 2005 also provides a graphical editing view of XML documents in Altova's popular Authentic View, thus enabling users to enter data into an XML document as they would into a wordprocessor-type application. Authentic View is particularly useful in situations where:

- people not familiar to XML are called upon to enter data into an XML document, or where
- several users input data into, or view, a single document located on a server or shared resource.

In this section, we provide a brief overview of the **main features** of XMLSpy 2005. These features are described in more detail in the various interface view sections (Text View, Schema/WSDL View, Authentic View, etc) of this document and in the User Reference. Please note that this is not a comprehensive list of available features. It is intended to give you a broad idea of what is possible with XMLSpy 2005.

Edit XML documents in multiple editing formats
You can edit an XML document as plain text (Text View) or in a graphical WYSIWYG view (Authentic View). For XML Schemas, you can also use Schema/WSDL View, which is a graphical interface that greatly simplifies the creation of complex schemas. You can also switch among the various views to suit your convenience. The Browser View enables you to directly view XML documents associated with an XSLT stylesheet and HTML documents.

Well-formedness checking and built-in validator
All XML documents are checked for well-formedness when you change views or save the file. XML documents can also be validated if a schema (DTD or XML Schema) is associated with the XML document. Other types of documents, such as DTDs, are also checked for errors in syntax and structure.

Intelligent Editing
If a schema is associated with an XML document, the auto-completion feature of Text View provides valuable editing help. As you type, pop-up menus containing the elements, attributes, and enumerated attribute values allowed at the cursor point appear. Additionally, the correct closing tags are automatically inserted when you complete opening tags, and attributes selected from the pop-up menu are inserted with opening and closing quotes. You can also enable XMLSpy 2005 to automatically insert mandatory elements and/or attributes when an element is inserted. Further, each view has a set of Entry Helpers that enable you to insert document components or specify the properties of the component selected in the Main Window.

Schema editing and management
You can create XML Schemas quickly and easily in the graphical Schema/WSDL View. This takes away much of the difficulty of knowing XML Schema structures, syntax, and design principles. You can also create DTDs that can be checked for correct syntax, plus convert between Schemas and DTDs.

Built-in XSLT 1.0 and XSLT 2.0 processors

The built-in XSLT 1.0 and XSLT 2.0 processors are compliant with the relevant W3C drafts. They enable you to transform XML documents directly from within the IDE using either XSLT 1.0 or XSLT 2.0 stylesheets, and to debug XSLT stylesheets using the XSLT Debugger.

Built-in XQuery 1.0 processor

The built-in XQuery 1.0 processor is compliant with the XQuery 1.0 W3C Working Draft of 23 July 2004. It enables you to execute and debug XQuery documents directly from within the IDE.

Transformations of XML documents

XML documents can be transformed directly from within the IDE, either with the internal (built-in) XSLT processor or with any external XSLT processor. To generate PDF from within the IDE, you can specify an external FO processor, and also transform XML to PDF with a single-click after specifying the XSLT stylesheet to use. Furthermore, parameter values can be passed to the XSLT transformation from the IDE itself.

Authentic View

Authentic View is a graphical view of an XML document in XMLSpy 2005. Users can enter data into the XML document as they would in a wordprocessor. The StyleVision Power Stylesheet that is used to specify the formatting of the XML document in Authentic View and how the data is input is created in Altova's StyleVision 2005 product. Note that Authentic View is also available in Altova's Authentic 2005 Desktop Edition, which is currently available free of charge.

2.3 User interface

XMLSpy 2005 has a graphical user interface (shown below), which is organized into three broad parts:

- **Info Window**, which displays meta information about the document item being currently edited.
- **Main Window**, where the document you edit appears. The number of document views available in the Main Window depends on the type of document being edited. You can switch between views whenever you want.
- **Entry Helper Windows**, which vary according to the type of document being viewed and the view selected in the Main Window. The Entry Helpers help you to graphically edit your document.

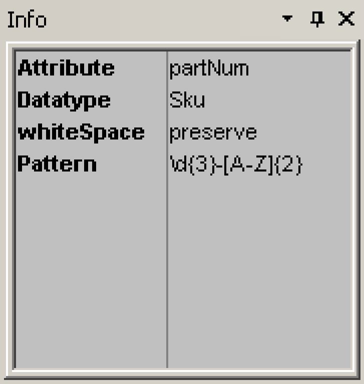
These windows can be docked under a menu bar and toolbar (*see screenshot below*), or they can be freely arranged under the menu bar and toolbar. Their positions and sizes can be changed by dragging and re-sizing them, as well as by using the toggle on/off commands in the **Window** menu.

This section provides an introduction to these broad parts of the interface. Detailed descriptions of the various interface parts follow this section.

Note: The Project window is not available in the Home Edition.

2.3.1 Info Window

XMLSpy 2005 provides a handy information window that shows detailed information about the element or attribute in which the cursor is currently positioned.



Attribute	partNum
Datatype	Sku
whiteSpace	preserve
Pattern	\d{3}-[A-Z]{2}

This information is available in all editing views and is especially helpful when used in conjunction with the `xsd:annotation` feature.

Note: You can turn the display of the Info Window on or off with the menu option **Window | Info window**.

2.3.2 Main Window

The Main Window is where you view and edit all documents in XMLSpy 2005.

Managing multiple open files

- You can open and edit any number of XML documents at one time in XMLSpy 2005.
- Each open file in the Main Window is opened in its own Document Window, and each has a tab with its name in it at the bottom of the Main Window.
- When open files are cascaded, tiled, or minimized, a title bar with (i) the name of the file, and (ii) standard minimize, maximize, and close buttons is displayed.
- Open files can be maximized or minimized by clicking the **Maximize** or **Minimize** button, respectively, in the title bar of any open file.
- When you maximize one file, all open files are maximized.
- Open files can be cascaded or tiled using commands in the Window menu.
- To make a file active (in order to edit it), click the file's tab or any part of its window. Alternatively, in the **Window** menu, select the required file from the list of currently open files at the bottom of the menu.
- You can also activate open files in the sequence in which they were opened by using **Ctrl+Tab** or **Ctrl+F6**.

Accessing File commands quickly
To access File commands quickly (such as printing, closing, sending as an e-mail attachment), right-click the file's tab. This opens a context-menu with a selection of File commands.

Main Window Views
XMLSpy 2005 provides multiple views of your XML document. These views are either editing or browser views:

- **Text View:** An editing view with syntax-coloring for source-level work
- **Schema/WSDL View:** For viewing and editing XML Schemas
- **Authentic View:** For editing XML documents based on StyleVision Power Stylesheets
- **Browser View:** An integrated browser view that supports both CSS and XSL stylesheets.

To switch between document views in the Document Window, click on the appropriate view button at the bottom of the Document Window. Alternatively, use the commands in the View menu.

Note: You can customize the default view (that is, the Main Window view) for individual file extensions. To do this, go to the Tools **|** Options dialog, and make the required settings in the **File types** and **View** tabs.

2.3.3 Entry Helpers

XMLSpy 2005 has intelligent editing features that help you to create valid XML documents quickly. These features are organized into three palette-like windows we call Entry Helpers.

When you are editing a document, the Entry Helpers display structural editing options according to the current location of the cursor. The Entry Helpers get the required information from the underlying DTD, XML Schema, and/or StyleVision Power Stylesheet. If, for example, you are editing an XML data document, then elements, attributes, and entities that can be inserted at the current cursor position are displayed in the relevant Entry Helpers windows, as well as information about these.

The Entry Helper windows have a XMLSpy 2005 prefix in Visual Studio .NET.

Entry Helpers in different views
What Entry Helpers are displayed depend upon the view. The different sets of Entry Helpers are categorized as follows, according to the views available in your Altova product:

- Text View: Elements, Attributes, and Entities Entry Helpers
- Schema Design View: Component Navigator, and Details and Facets Entry Helpers
- Authentic View: Elements, Attributes, and Entities Entry Helpers

The Entry Helpers for each view are described in the section about that view (in the following sections of this documentation).

Note: You can turn the display of Entry Helpers on or off with the menu option **Window | Entry Helpers**.

2.3.4 Menu Bar and Toolbar

Menu Bar
The menu bar contains the various application menus. The following conventions apply:

- If commands in a menu are **not** applicable in a view or at a particular location in the document, they are grayed out and disabled.
- Some menu commands pop up a submenu with a list of additional options. Menu commands with submenus are indicated with a right-pointing arrowhead to the right of the command name.
- Some menu commands pop up a dialog that prompts you for further information required to carry out the selected command. Such commands are indicated with an ellipsis (...) after the name of the command.
- To access a menu command, click the menu name and then the command. If a submenu is indicated for a menu item, the submenu opens when you mouseover the menu item. Click the required sub-menu item.
- A menu can be opened from the keyboard by pressing the appropriate key combination. The key combination for each menu is **Alt+KEY**, where KEY is the underlined letter in the menu name. For example, the key combination for the **File** menu is **Alt+F**.
- A menu command (that is, a command in a menu) can be selected by sequentially selecting (i) the menu with its key combination (see previous point), and then (ii) the key combination for the specific command (**Alt+KEY**, where KEY is the underlined letter in the command name). For example, to create a new file (**File | New**), press **Alt+F** and then **Alt+N**.
- Some menu commands can be selected **directly** by pressing a special **shortcut** key or key combination (**Ctrl+KEY**). Commands which have shortcuts associated with them are indicated with the shortcut key or key combination listed to the right of the command. For example, you can use the shortcut key combination **Ctrl+N** to create a new file; the shorcut key **F8** to validate an XML file.

Toolbar
The toolbar contains buttons that are shortcuts for commands found in the menus. The name of the command appears when you place your mouse pointer over the button. To execute the command, click the button.

Toolbar buttons are arranged in groups. In the **Tools | Customize | Toolbars** dialog, you can specify which toolbar groups are to be displayed. In the GUI, you can also drag toolbar groups to alternative locations by clicking and dragging a toolbar handle to the desired location.

XMLSpy 2005 Home Edition

Tutorials

3 Tutorials

The following tutorials will give you a good fundamental understanding of how XMLSpy 2005 works and of its main features.

- The XMLSpy 2005 Tutorial shows you how to create XML Schemas using the features available in XMLSpy 2005, how to create XML documents based on a schema, and how to transform your XML document using an XSLT stylesheet.
- The Authentic View Tutorial shows you how to use Altova's Authentic View, which is a graphical display and data-entry template for XML documents. Authentic View is available in a standalone product (Authentic 2005 Desktop Edition) and as a view in XMLSpy 2005.

3.1 XMLSpy Tutorial

The XMLSpy 2005 tutorial gives you a hands-on introduction to XMLSpy 2005. By the time you have worked through it, you will have created an XML Schema, an XML file based on this schema, and will have transformed the XML file using an XSLT stylesheet. You will also have learned how to use some of the fundamental and most commonly used features of XMLSpy 2005.

3.1.1 XMLSpy Home Edition Tutorial

This tutorial gives a short overview of XML, and takes you through several tasks which provide an overview of how to use XMLSpy 2005 to its fullest.

You will learn how to:

- Create a **simple schema** from scratch
- **Generalize** the schema using simple and complex types
- **Validate** the XML document against its schema
- **Transform** the XML document into HTML using XSLT, and view the result in the Browser view

XMLSpy 2005 installation and configuration
This tutorial assumes that you have successfully installed XMLSpy 2005 on your computer as a registered user, or you have received a free evaluation key-code for XMLSpy 2005.

The evaluation version of XMLSpy 2005 is fully functional but time-limited to 30 days. You can request a regular license from our secure web server or through any one of our resellers.

Tutorial example files
The tutorial files are available in the **..\Examples\Tutorial** folder.

The **Examples folder** contains various XML files for you to experiment with, while the **Tutorial** folder contains all the files used in this tutorial.

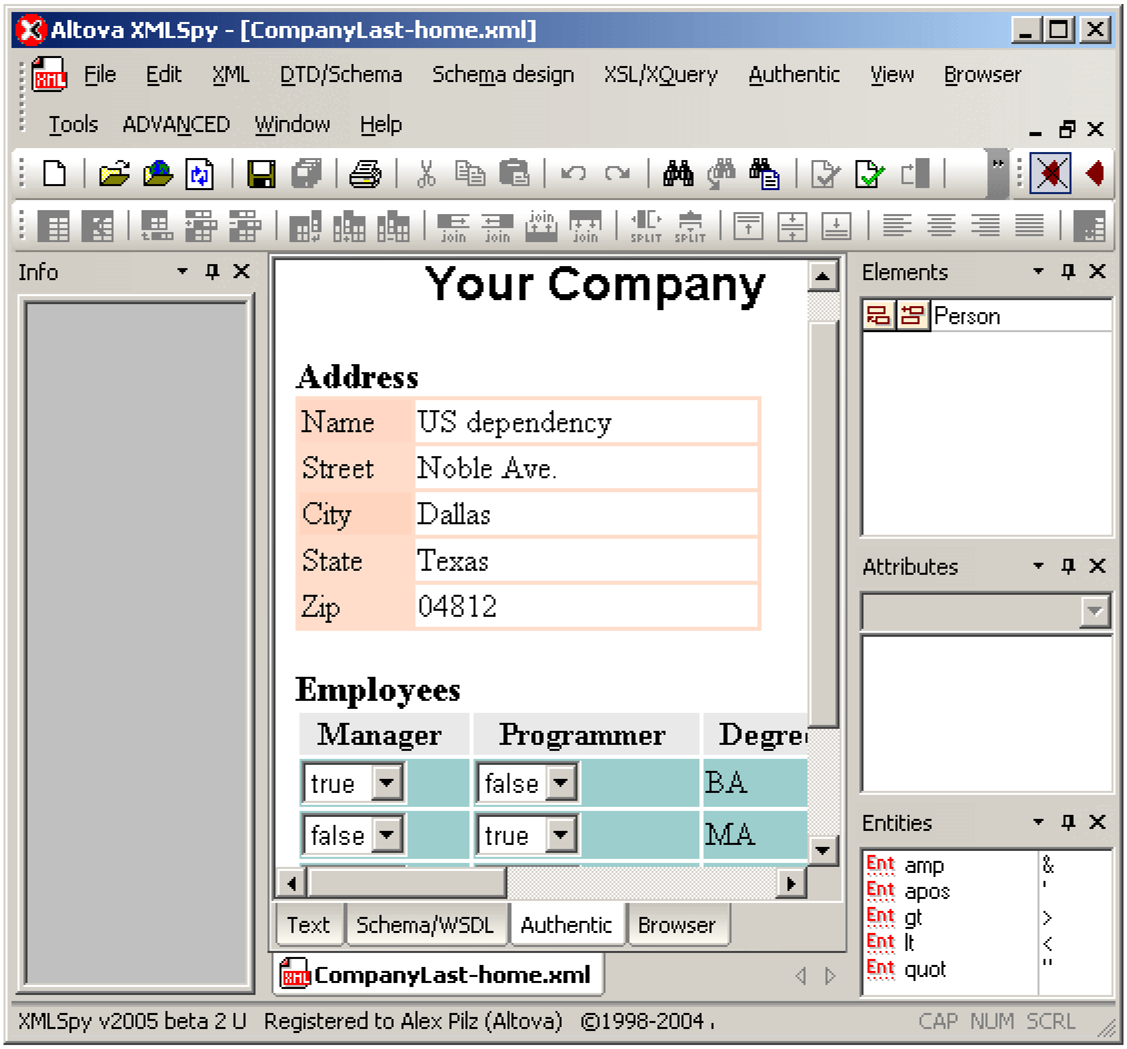
The **Template folder** contains all the XML template files that are used whenever you select the menu option **File | New**. These files supply the necessary data (namespaces and XML declarations) for you to start working with the respective XML document immediately.

The interface

XMLSpy 2005 provides several windows that show various aspects of your XML document:

- The left area consists of the **Info** window.
- The central area, called **Main** window, is where you edit and view all types of XML documents.

  You can choose from different views: Text view, Schema/WSDL view, Authentic View or Browser view.
- The right area contains the three **Entry helper** windows which allow you to insert or append: elements, attributes, and entities.


Altova XMLSpy - [CompanyLast-home.xml]
File Edit XML DTD/Schema Schema design XSL/XQuery Authentic View Browser
Tools ADVANCED Window Help
Info
Elements
Person
Attributes
Entities
Ent amp &
Ent apos '
Ent gt >
Ent lt <
Ent quot "
Text Schema/WSDL Authentic Browser
CompanyLast-home.xml
XMLSpy v2005 beta 2 U Registered to Alex Pilz (Altova) ©1998-2004 ,
CAP NUM SCRL

Your Company

Address

Name	US dependency
Street	Noble Ave.
City	Dallas
State	Texas
Zip	04812

Employees

Manager	Programmer	Degre
true	false	BA
false	true	MA

Creating a schema from scratch

A Schema describes what one or more XML documents can look like,
and defines:

- The elements the document contains, and the order in which they appear
- The element content, and element attributes if any

The purpose of a schema is to allow machine validation of document structure. Instead of using the syntax of XML 1.0 DTD declarations, schema definitions use XML element syntax. A correct XML schema definition is, therefore, a well-formed XML document.

Goal of this section:
The goal of this section is to **create a simple schema** describing a company and its employees. The company is to consist of an **address** and an unlimited number of **persons**.

This will be achieved by:

- **Adding** elements to the schema
- Defining element **sequences**
- Adding **sub-elements** to an element (child elements)
- Creating elements using **drag and drop**

- Making an element **optional**
- Defining an element **facet**

Functions (and their icons) in this section:

File | New, creates a new XML instance file.

Schema design | Display diagram displays the content model of the selected global component in the top part of the main window. To display the content model of a component, click the "Display diagram" icon located to the left of each component in the "Display all globals" view of the Schema overview. The "Display diagram" function toggles with the "Display all globals" function.

Schema design | Display all globals displays all global components of the schema in the top part of the main window. The "Display all globals" function toggles with the "Display diagram" function.

TAB Takes you to the next field and automatically opens a drop-down list if one exists.

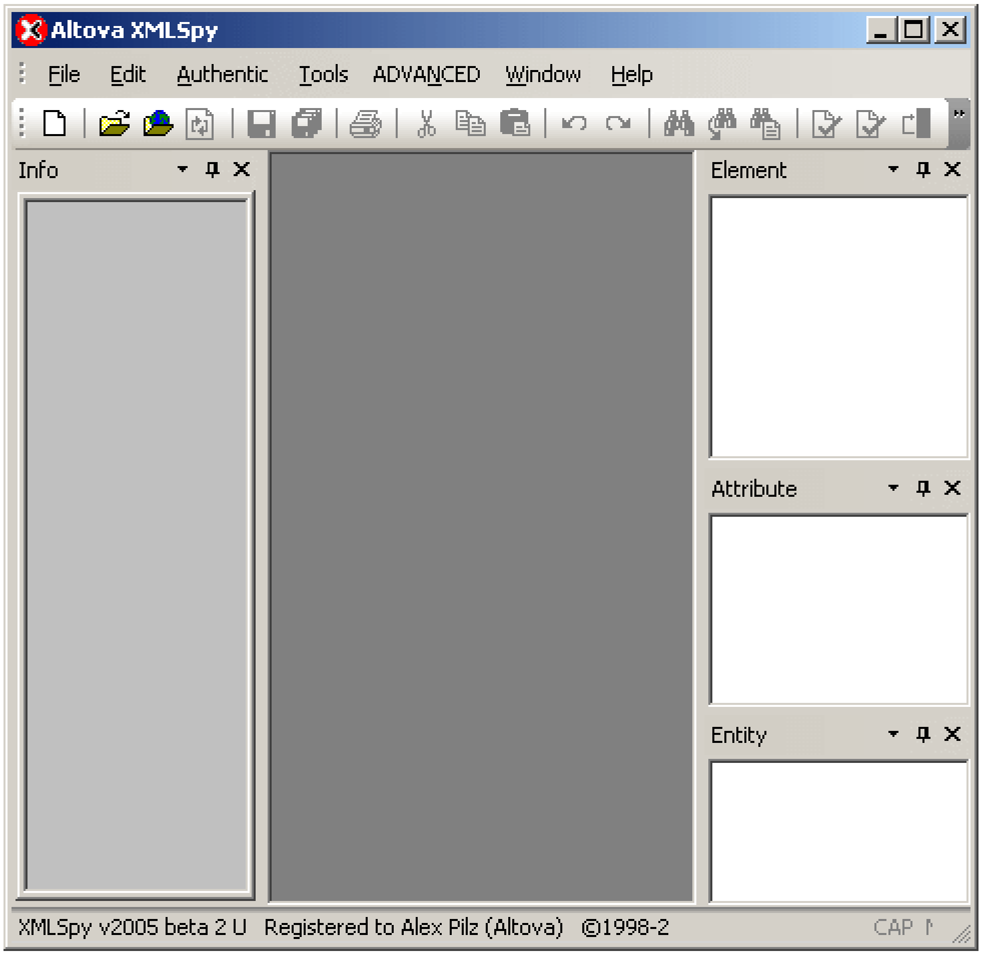
CTRL **+ Drag&Drop**, enables you to copy existing elements.

Append icon, allows you to append an element to the schema.

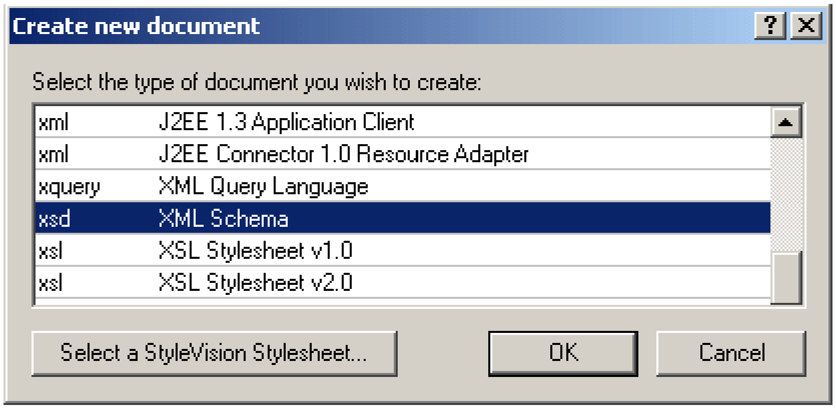
Creating a new Schema file

To create a new schema file:

1. Start XMLSpy 2005 by double clicking on the XMLSpy 2005 icon.
   You are presented with an empty environment. There are no XML documents in the main window.


Altova XMLSpy
File Edit Authentic Tools ADVANCED Window Help
Info
Element
Attribute
Entity
XMLSpy v2005 beta 2 U Registered to Alex Pilz (Altova) ©1998-2
CAP

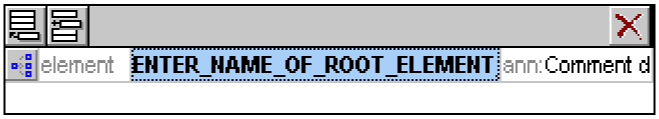
2. Select the menu option **File | New** and select the .xsd W3C XML Schema entry from the dialog and confirm with OK.



Select the type of document you wish to create:

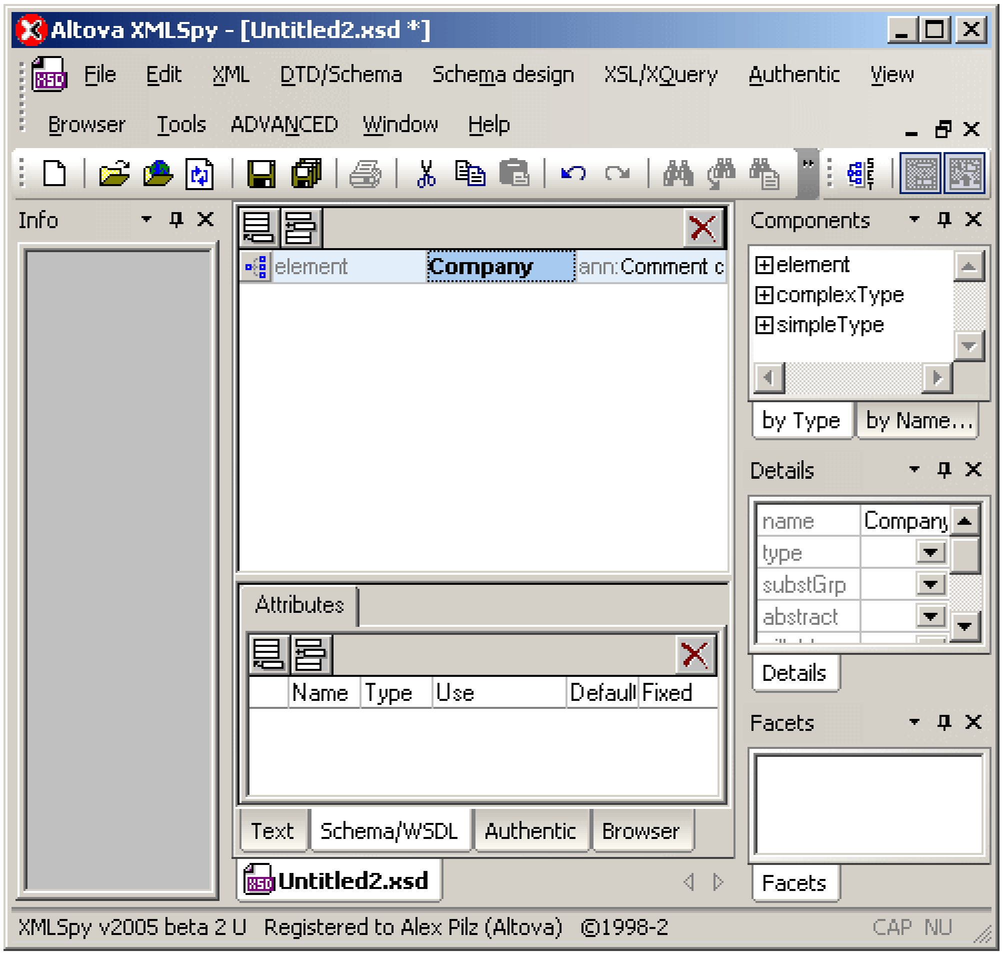
xml	J2EE 1.3 Application Client
xml	J2EE Connector 1.0 Resource Adapter
xquery	XML Query Language
xsd	XML Schema
xsl	XSL Stylesheet v1.0
xsl	XSL Stylesheet v2.0

An empty schema file appears in the main window. You are prompted to enter the name of the root element.



3. Click in the highlighted field and enter "Company", confirm with Enter. Company is now the "root" element of this schema and is automatically a "global element" as well.

   This view is the **Schema overview** and displays the **global components** in the top window and the **attributes** of the currently selected component, in the lower one.



element Company ann:Comment c

Name	Type	Use	Default	Fixed

name	Company
type	
substGrp	
abstract	

XMLSpy v2005 beta 2 U Registered to Alex Pilz (Altova) ©1998-2

The top entry helper window, the **Component Navigator**, displays Company in the "Elm" tab. The entries in these tabs can be used to navigate your schema by double clicking on them.

4. Click the menu option **File | Save as**, and name your schema (**AddressFirst** for example).

Defining your own namespace:

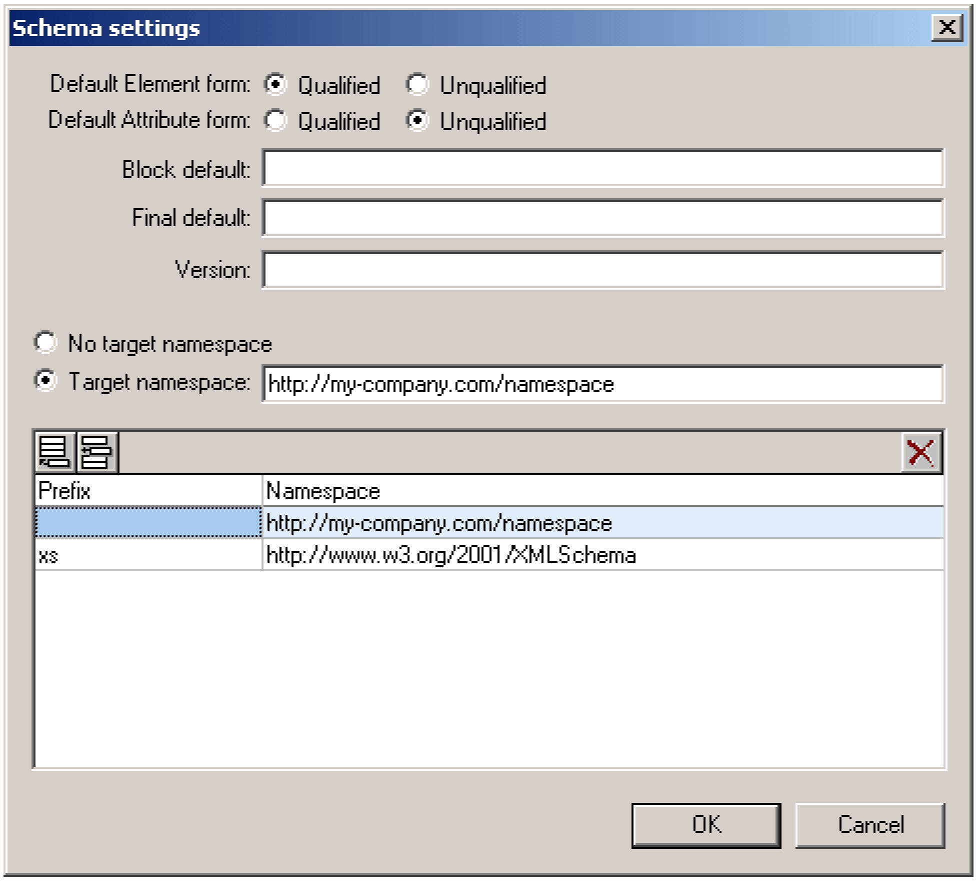
1. Select the menu option **Schema Design | Schema settings.**
2. Click the Target namespace radio button, and enter **http://my-company.com/namespace.**

Please note:

The namespace defined above will also be used in two related files:

- the **XML** file you will create later in this tutorial and
- in the **XSLT stylesheet** that will be used to transform the XML to HTML.

The namespace must be **identical** in all three files (Schema, XML, and XSL). So if you enter any other namespace than that given above, please ensure that the corresponding namespace entries in the XML and XSLT files match.



Schema settings

Default Element form: ● Qualified ○ Unqualified

Default Attribute form: ○ Qualified ● Unqualified

Block default:

Final default:

Version:

○ No target namespace

● Target namespace: http://my-company.com/namespace

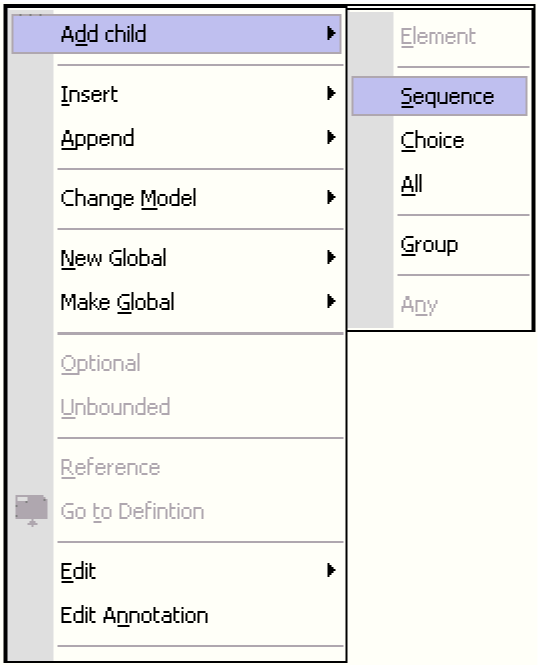
Prefix	Namespace
	http://my-company.com/namespace
xs	http://www.w3.org/2001/XMLSchema

3. Confirm with the OK button.

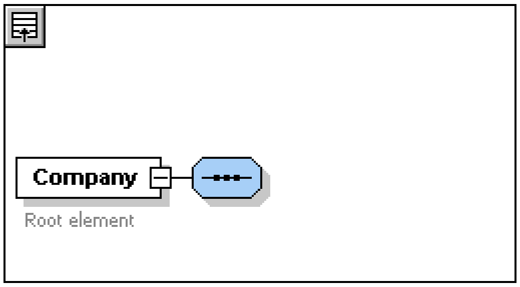
Adding elements to a schema

To add elements to a schema:

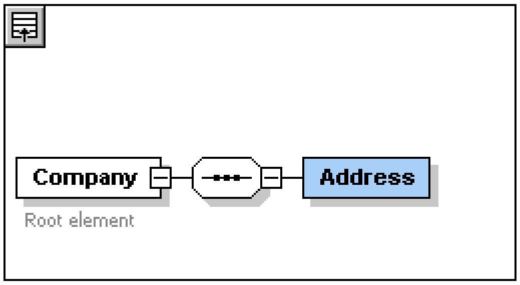
1. Click the component icon next to the **Company** element, in the main window, to display the content model (or double click on the Company entry in the Component Navigator).
   The text below the company element is annotation text. Double click the text if you want to edit it. (shortened to "Root element" here.)
2. Right click the Company element to open the context menu, and select **Add Child | Sequence**.



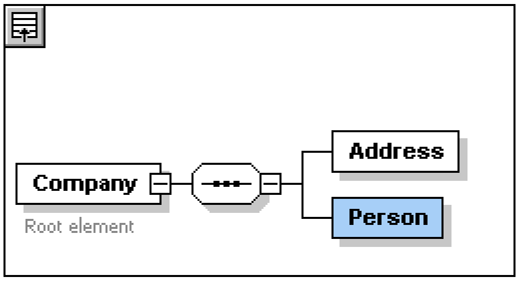
This inserts the **Sequence compositor**, and defines that the following elements must appear in the same sequence (in the XML document).



3. Right click the Sequence compositor and select **Add Child | Element**.
4. Enter "Address" as the name of the element, and confirm with the Enter key.

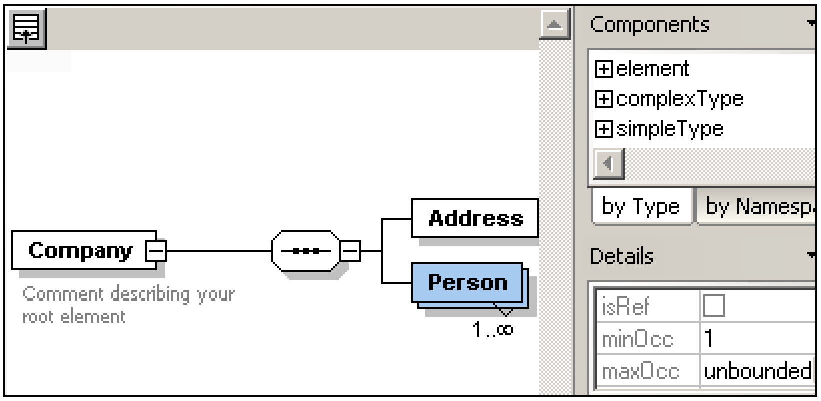


5. Right click the Sequence compositor again, select **Add Child | Element**, and enter "Person" as the name of the element.



We have now defined a schema which allows for one address and one person per company. As this is too restrictive, we want to make sure that we can include as many persons per company as necessary.

6. Right click the Person element, and select **Unbounded** from the context menu. The Person element changes at this point, showing the range in which it can occur, in this case 1 to infinity.



isRef	□
minOcc	1
maxOcc	unbounded

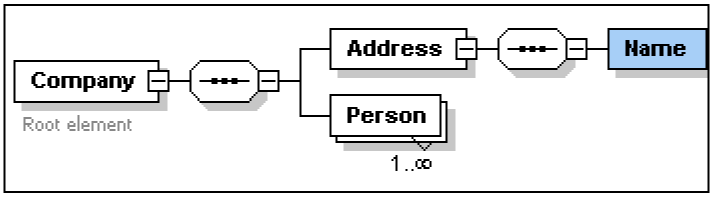
Please note:

You can also edit the **minOcc** and **maxOcc** fields in the Details entry helper directly.

We will now add the sub-elements which define the address structure.

To add sub-elements to an element:

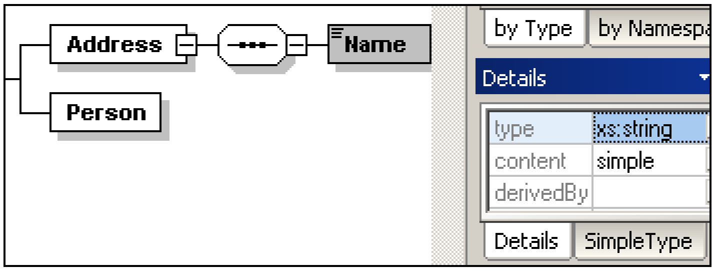
1. Right click the **Address** element to open the context menu, and select **Add Child | Sequence**.
2. Right click the **Sequence** compositor, and select **Add Child | Element**. Enter "Name" as the element name.



Defining element parameters:

At this point we want to define that the Name element is to occur only once, and contain only textual data.

1. Click the Name element, if not currently selected.
2. Click on the **type** combo box of the middle entry helper, and select the entry **xs:string** from the drop down list.

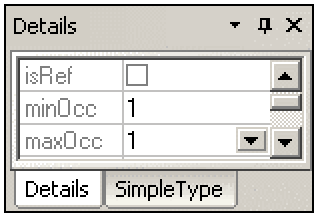


type	xs:string
content	simple
derivedBy	

This entry helper is called "**Details**" in the Schema/WSDL view, and provides information on the currently selected element. **All data can be edited directly in the Details window!**

An icon appears in the top left of the element **Name**, indicating that this element contains text.

Both "minOcc" and "maxOcc" fields contain 1, showing that there is only one occurrence of this element (this is the default setting when creating a new element).



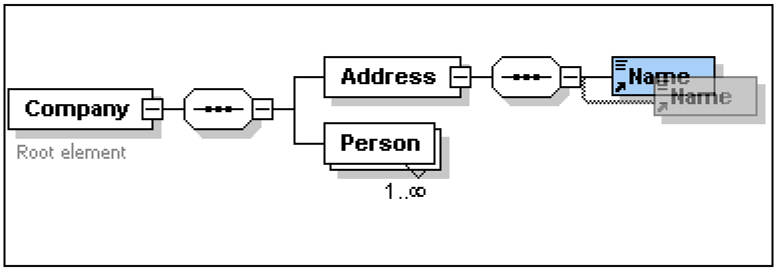
isRef	□
minOcc	1
maxOcc	1

Adding elements using drag and drop

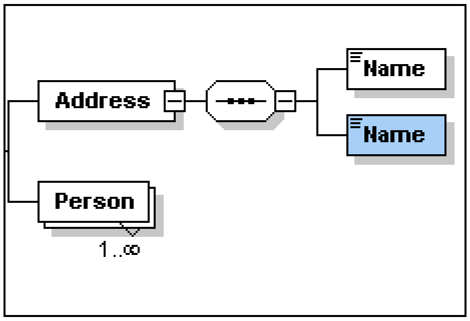
To add elements using drag and drop:

There is a quicker method of adding new elements to a schema, which avoids multiple menu commands:

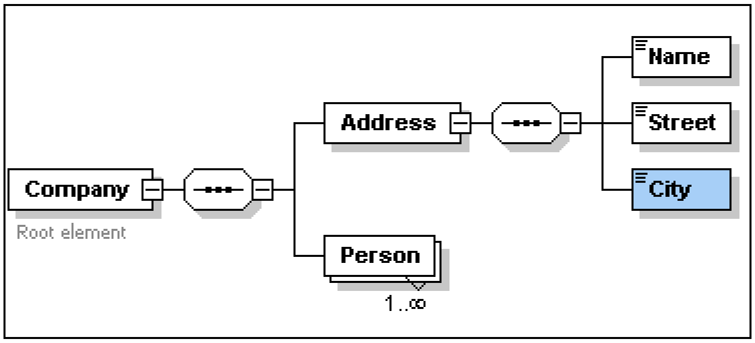
1. Click the Name element, hold down the **CTRL** key, and **drag** "slightly" with the mouse. A small "plus" icon appears as well as a copy of the element, showing that you are about to copy the element.



2. Release the mouse button to create the new element. If the new element appears somewhere else, just drag it near to the Name element and drop it there. This method creates an element of the same type, with the same settings as the one copied.



3. Type "Street" to change the element name.
4. Use the same method to create a third element, "City".
   The content model should now look like this:



Completing the basic schema

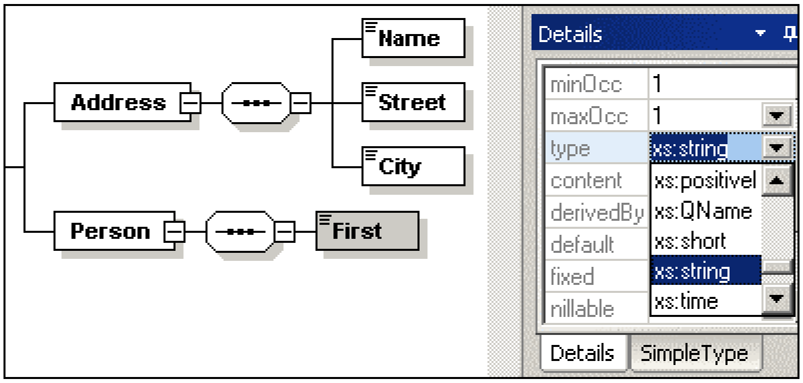
At this point we want to add those sub elements to the Person element, which make up the personal data. All these elements will be **simple types** (with **simple content** models).

Person sub-elements: First, Last, Title, PhoneExt, and Email.

Requirements:

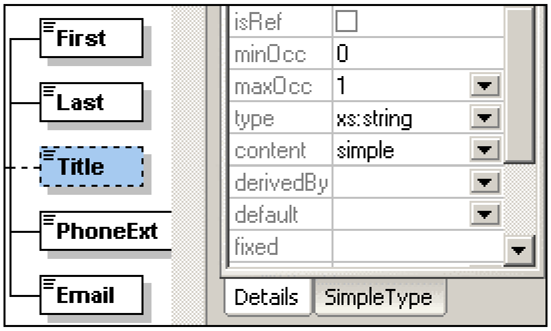
Title element:	should be **optional**
PhoneExt:	should be an **integer** and limited to **2 digits**

1. Right click the Person element to open the context menu, and select **Add Child | Sequence**. This inserts the Sequence compositor.
2. Right click the **Sequence** compositor, and select **Add Child | Element**.
3. Enter "**First**" as the name of the element. Press 'Enter'. Go to the Details window and click on the down-arrow in the **type** row.



minOcc	1
maxOcc	1
type	xs:string
content	
derivedBy	
default	
fixed	
nillable	

4. Select the **xs:string** entry from the drop down list.
5. Use the drag and drop method to create **four more elements**, and name them: Last, Title, PhoneExt, and Email respectively.



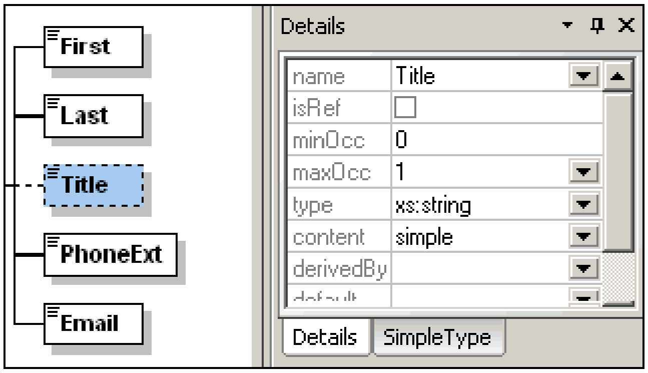
isRef	□
minOcc	0
maxOcc	1
type	xs:string
content	simple
derivedBy	
default	
fixed	

Please note:

You can select multiple elements by holding down the CTRL key, and clicking each one.

To make an element optional:

1. Right click the **Title** element, and select **Optional** from the context menu. The solid element frame changes to a dashed one; this is the visual display that an element is optional.

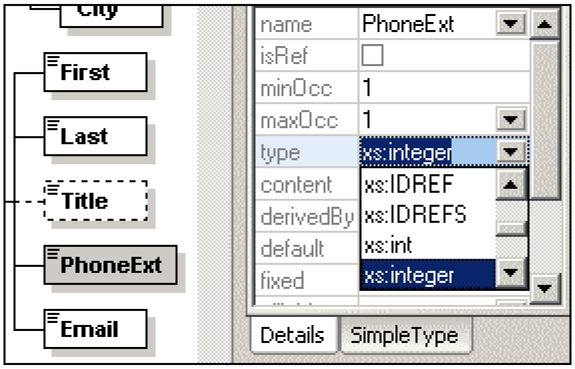


name	Title
isRef	□
minOcc	0
maxOcc	1
type	xs:string
content	simple
derivedBy	

The "Details" fields have also been updated **minOcc=0** and **maxOcc=1**.

To limit the content of an element (Facets):

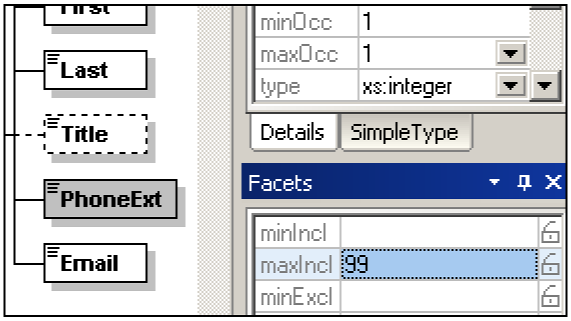
1. Click the **type** field of the PhoneExt **element**, and select (or enter) the **xs:integer** entry from the drop down list.



name	PhoneExt
isRef	□
minOcc	1
maxOcc	1
type	xs:integer
content	xs:IDREF
derivedBy	xs:IDREFS
default	xs:int
fixed	xs:integer

The items in the Facets tab (in the lowest entry helper) change at this point.

2. Double click in the "**maxIncl**" field of the Facets tab (in the lowest entry helper) and enter **99**, confirm with Enter.



minOcc	1
maxOcc	1
type	xs:integer

Facets

minIncl	
maxIncl	99
minExcl	

This defines that all phone extensions up to, and including 99, are valid.

3. Select the menu option **File | Save** to save the changes to the schema.

Please note:

- Selecting a predefined simple type "text" (i.e. xs:string, xs:date etc.) for an element, automatically changes the content model to: content = simple, in the Details entry helper.
- Adding a compositor to an element (selection, choice or all), automatically changes the content model to: content = complex, in the Details entry helper.
- This schema is available as '**AddressFirst.xsd**' in the ..\Tutorial folder.

Making schema components reusable

Goal of this section:

To create generic **schema** components which can be reused by other elements.

This will be achieved by:

- Creating a global **AddressType** component, which will be the basis for specific country addresses (a complex type)
- Creating two specific address templates for UK-, and US Adresses by **extending** the global address element (extend the complex type)
- Creating a global US-State element, by **restriction** (simpleType)
- Creating a global person element by **reference**

- Defining person **attributes** that supply information about the persons position in the company
- **Limiting** the **attribute contents** to a predefined set of attribute values (enumeration)

Functions (and their icons) in this section:

Schema design | Display all globals, takes you back to the schema overview.

Append icon, allows you to append an element or attribute to a schema.

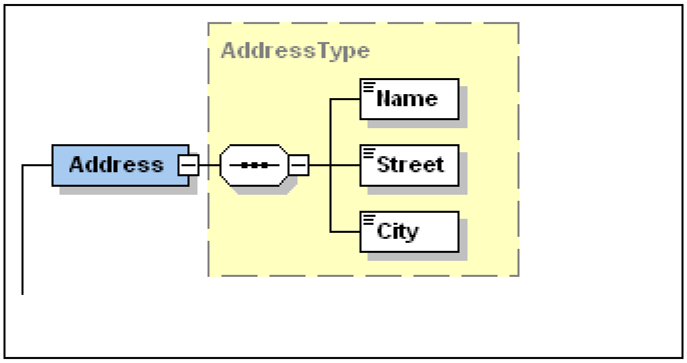
Schema design | Display diagram, the component icon displays the content model of the active global component in the schema overview.

Globals, extending simple and complex types

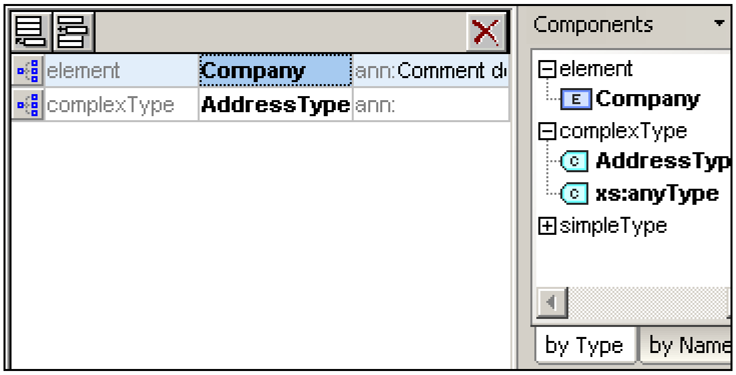
Having defined an element, you may then realize that you want to reuse it somewhere else in your schema. In XMLSpy 2005 this is achieved by creating a **global component**.

To create a global component:

1. Right click the Address element, and select **Make Global | Complex type**. The Address elements appear in a yellow box.



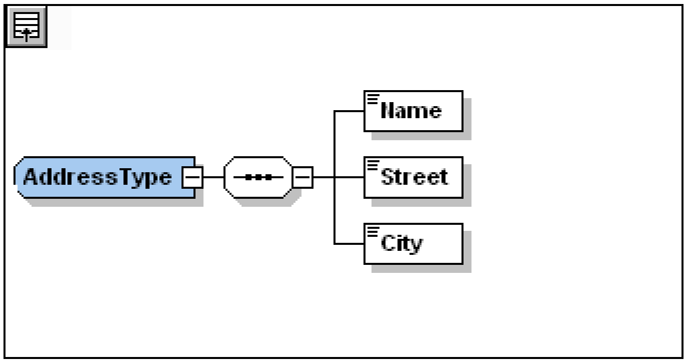
2. Click on the "Display all Globals" icon. The schema overview now displays two global components: the Company element and the complexType "AddressType".



element	Company	ann:Comment d
complexType	AddressType	ann:

Click the **Element** and **complexType** entries in the Components entry helper, to see the respective schema constucts.

3. Click on the AddressType component icon, to see the content model.

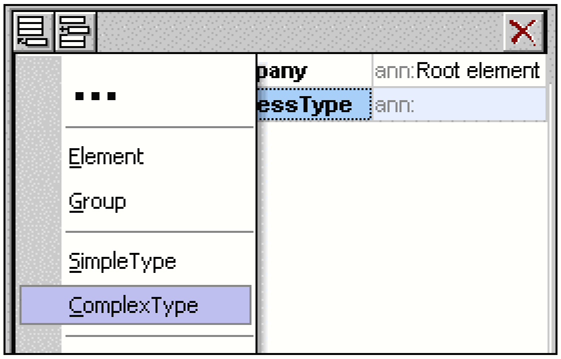


4. Click the "Display all Globals" icon to return to the schema overview.

Extending a "complex type" definition
We now want to use the global AddressType component, to create two kinds of country specific addresses. For this purpose we will define a **new** complex type **based** on the AddressType component.

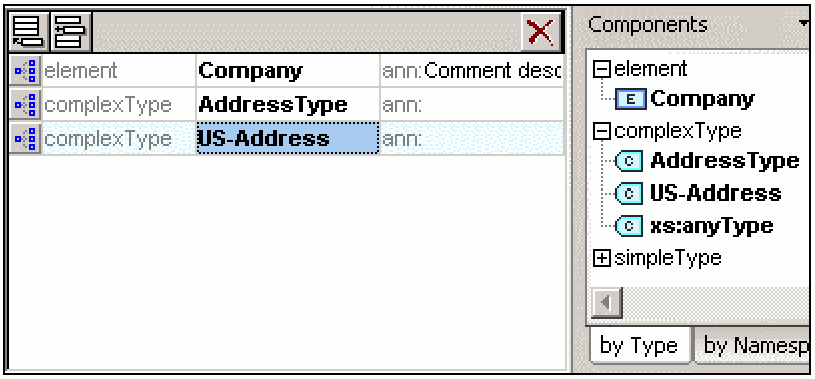
To extend a "complex type" definition:

1. Switch to the schema overview, if not already visible (Display all globals).
2. Click the **Append** icon, at the top left of the component window.
3. Select **ComplexType** from the context menu.



A new line appears in the component list, and the cursor is set for you to enter the component name.

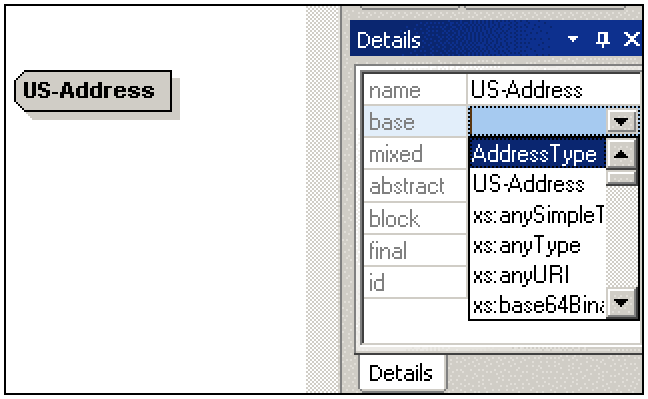
4. Enter "US-Address" and confirm with Enter. (If you enter US-Address with a blank space instead of a hyphen character "-", the element name will appear in **red,** signalling an illegal character.)



element	Company	ann:Comment desc
complexType	AddressType	ann:
complexType	US-Address	ann:

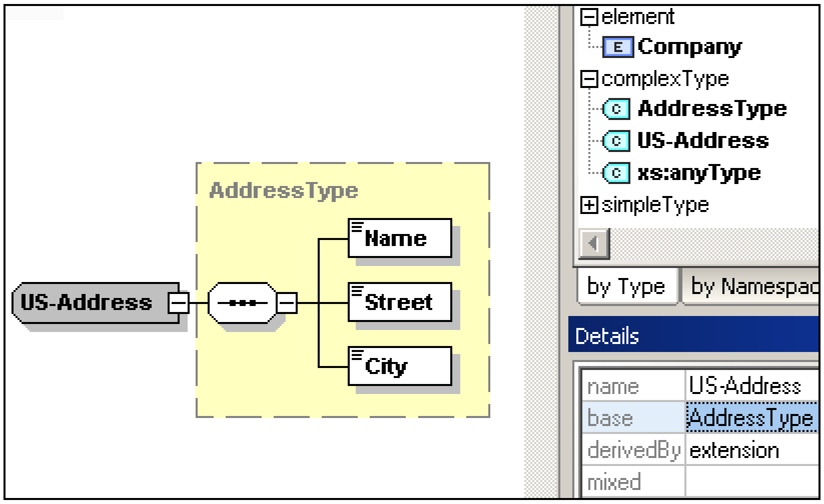
5. Click the **US-Address** component icon to see the content model.
6. Click the "**base**" combo box in the Details entry helper, and select the "AddressType"

entry.



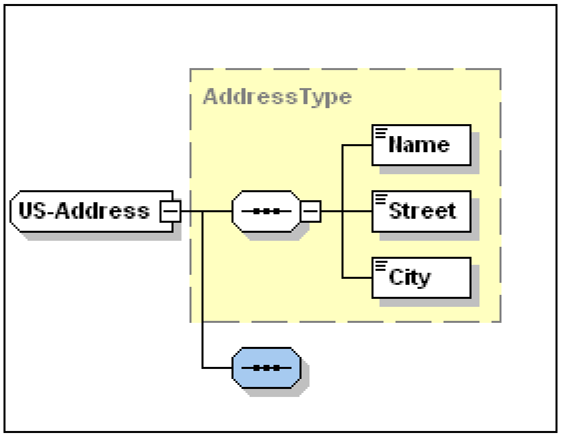
name	US-Address
base	
mixed	
abstract	
block	
final	
id	

The content model view changes immediately and displays the previously defined generic address.



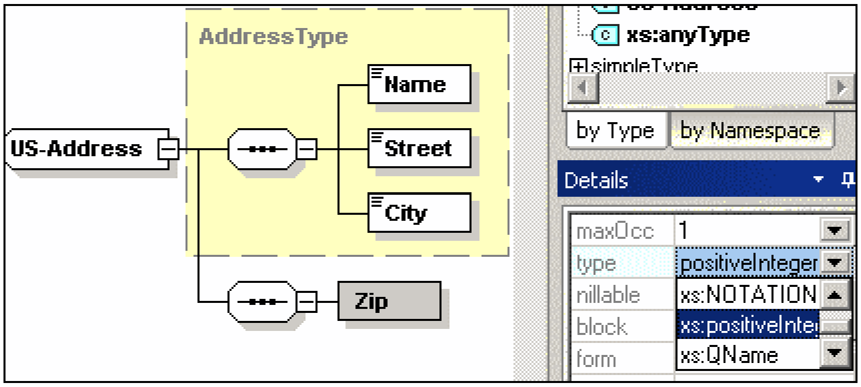
name	US-Address
base	AddressType
derivedBy	extension
mixed	

7. Right click the US-Address **element**, and select **Add Child | Sequence**.
   A new sequence compositor is displayed **outside** of the AddressType box. This is a visual indication that this is an **extension** to the element.



8. Right click the new **sequence** compositor, and select **Add Child | Element**.
9. Name the element "Zip", and press 'Enter'.
10. Select (or enter) **xs:positiveInteger** from the "type" field combo box, and confirm with

Enter.



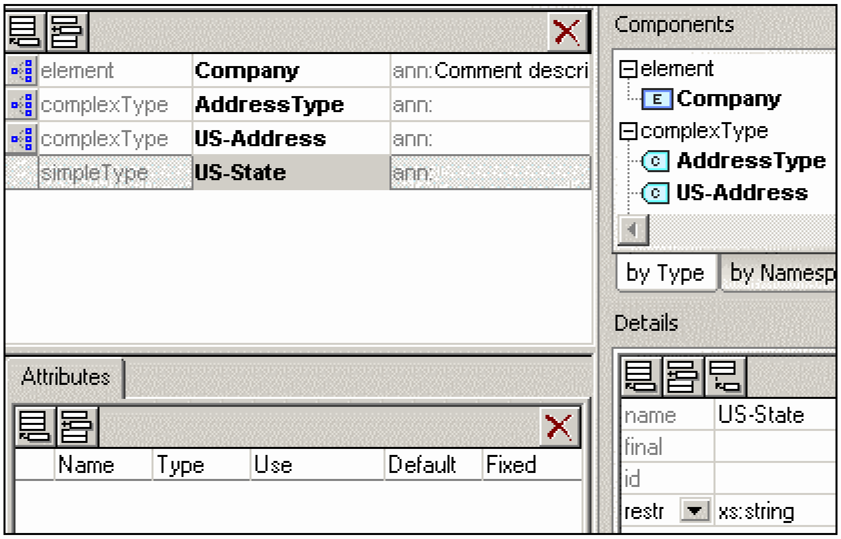
maxOcc	1
type	positiveInteger
nillable	xs:NOTATION
block	xs:positiveInte
form	xs:QName

Creating reusable "simple type" elements

Simple type elements can also be made generic. In this case we want to make the State element reusable, so that an abbreviated version could also be included in address labels at a later time (GA for Georgia, for example).

To create reusable "simple type" elements:

1. Switch to the Schema overview (Display all Globals).
2. Click the **append** icon, select SimpleType, and enter "US-State" as the element name (Enter to confirm).
3. Select **xs:string** in the "**restr**." value field of the Details entry helper.
   This completes the definition. This element can now be used in the US-Address definition.

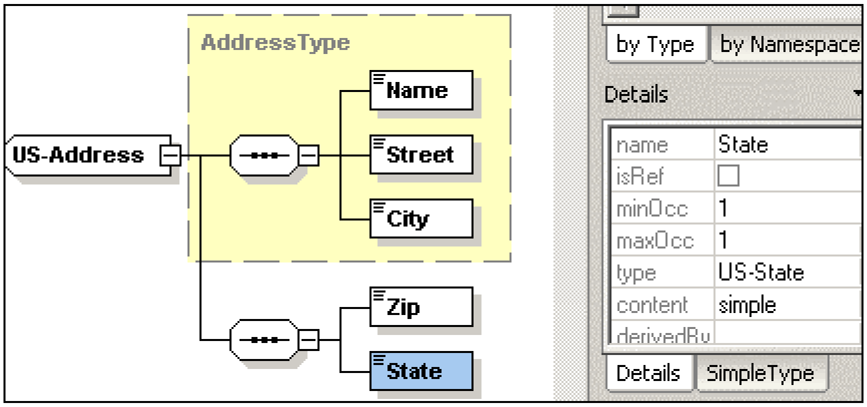


element	Company	ann:Comment descri
complexType	AddressType	ann:
complexType	US-Address	ann:
simpleType	US-State	ann:

Name	Type	Use	Default	Fixed

name	US-State
final	
id	
restr	xs:string

4. Click the US-Address component icon , then right click the lower sequence compositor and select **Add Child | Element**.
5. Enter "State" for the element name, move the pointer to the "Details"window.
6. Select (or enter) "US-State" from the "type" combo box (click Enter to confirm).



name	State
isRef	□
minOcc	1
maxOcc	1
type	US-State
content	simple
derivedBy	

Please note:

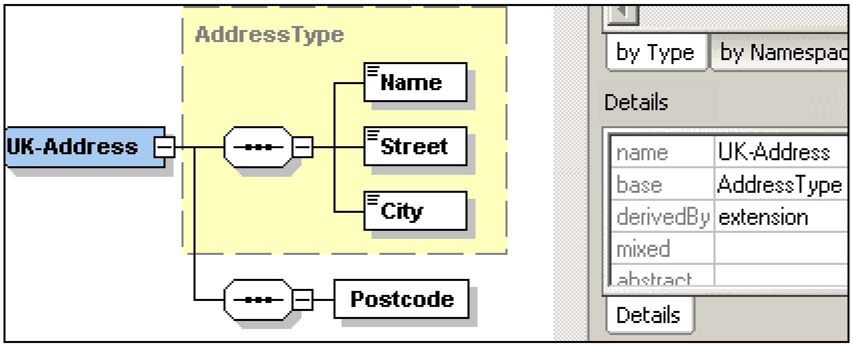
Global simple types can only be created from the schema overview.

Creating the second Address template

Using the method described above, define the global complex type "**UK-Address**".

1. Create the global **complex type** "UK-Address", with the **base**="AddressType"
2. Add a new Postcode element to the content model of UK-Address.

   Your UK-Address content model should finally look like this:



name	UK-Address
base	AddressType
derivedBy	extension
mixed	
abstract	

Please note:

Global definitions (global elements, complex types, etc.) can be moved or copied to other schemas visible in the schema overview, using drag and drop. You can of course, reposition definitions in the currently open schema.

Right clicking a definition opens the context menu in which you can select the standard cut, copy, paste commands, to achieve the same thing. The drag and drop method also applies to attributes visible in the Attributes tab.

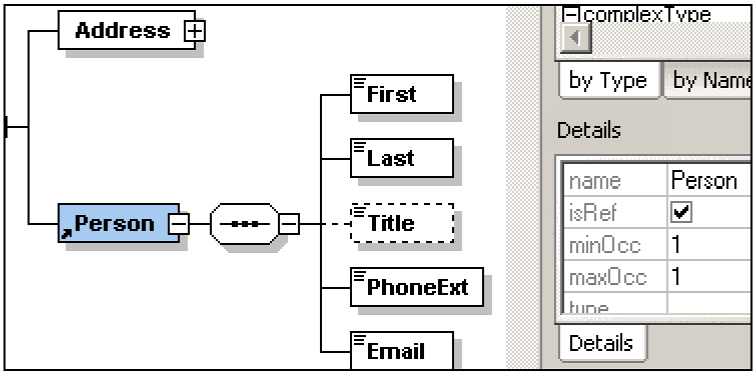
References, attributes and enumerations

To finish off the schema definition we will make the Person element global, define specific element attributes and limit the attribute selection.

To create a reference:

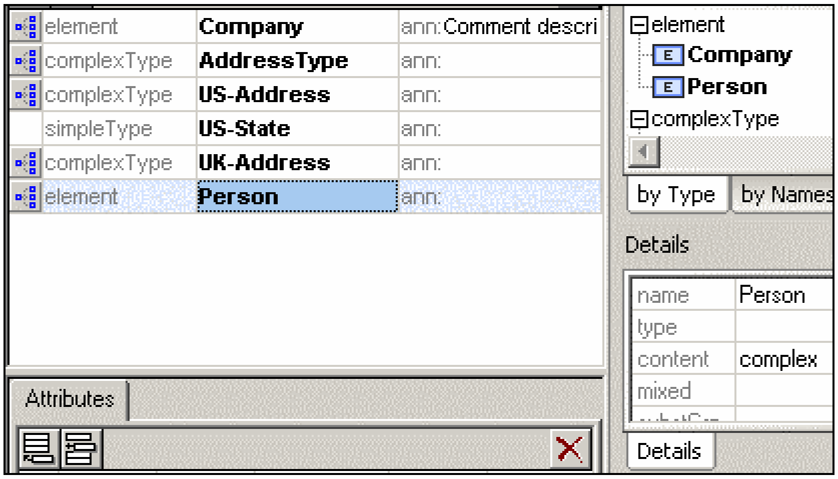
1. Switch to the Schema overview (Display all Globals).
2. Click on the component icon of the **Company** element.
3. Right click the Person element, and select **Make Global | Element**.

   A small "link" icon appears in the Person element, showing that this element now references the globally declared "Person" element. The "**isRef**" field in the Details entry helper is set active.



name	Person
isRef	☑
minOcc	1
maxOcc	1
type	

4. Click the "Display all Globals" icon to return to the schema overview.
The Person element is now also visible in the component list, as well as in the "Elm" tab of the Component navigator. Click the Elm tab to see the global elements.



element	Company	ann:Comment descri
complexType	AddressType	ann:
complexType	US-Address	ann:
simpleType	US-State	ann:
complexType	UK-Address	ann:
element	Person	ann:

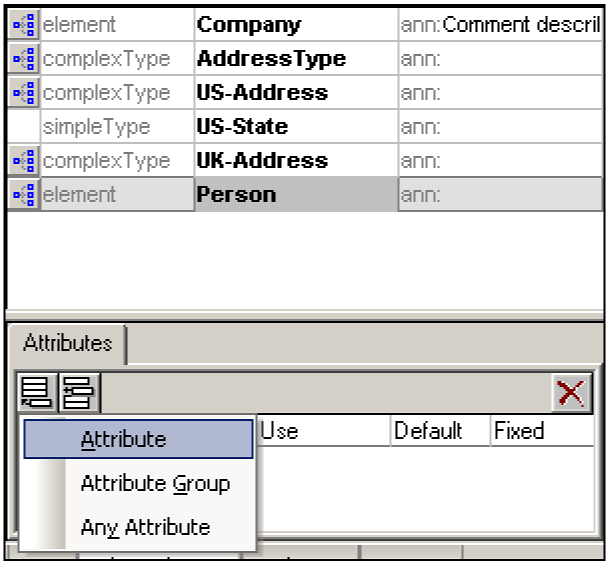
name	Person
type	
content	complex
mixed	

Please note:

Global declarations do not describe where an element is to be used in an XML document, they only describe what it contains. Global definitions have to be referenced from within a complex type, or another element, to determine their position in the XML document.

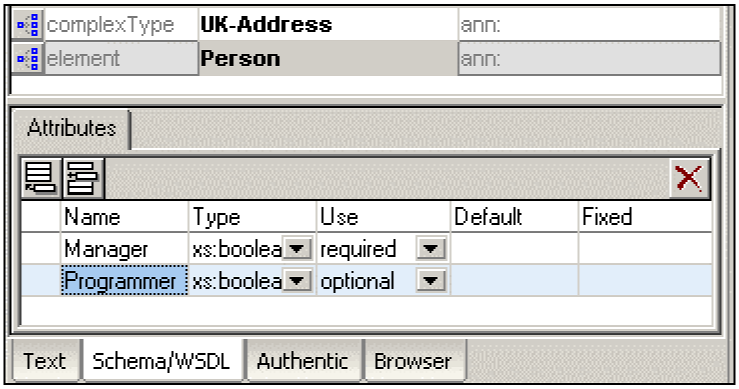
To define Element attributes:

1. Click the Person element to make it active.
2. Click the Append icon, in the top left of the **attribute** tab (the lower window of the schema overview), and select the "Attribute" entry.



element	**Company**	ann:Comment descril
complexType	**AddressType**	ann:
complexType	**US-Address**	ann:
simpleType	**US-State**	ann:
complexType	**UK-Address**	ann:
element	**Person**	ann:

3. Enter "**Manager**" as the attribute name in Name field.
4. Use the **Type** combo box to select "xs:boolean".
5. Use the **Use** combo box to select "required".



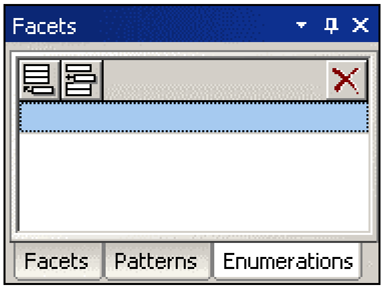
complexType	**UK-Address**	ann:
element	**Person**	ann:

Name	Type	Use	Default	Fixed
Manager	xs:boolea	required		
Programmer	xs:boolea	optional		

6. Use the same method to:
   Add a "Programmer" attribute in the Name field (type="xs:boolean), and set its Use to "optional".

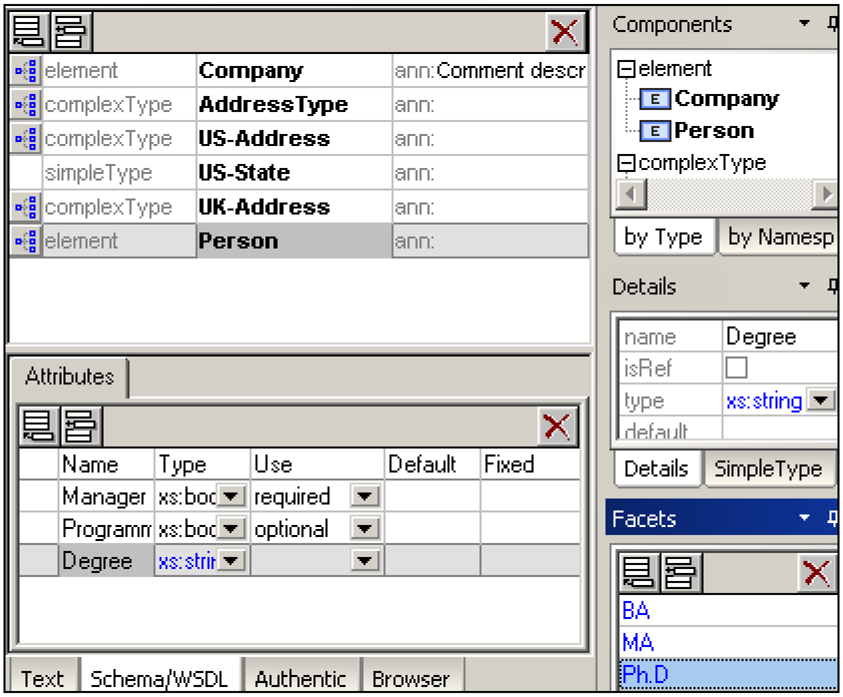
To limit the contents of an attribute (Enumerations):

1. Click the **Append** icon in the top left of the attribute window, and select the "**attribute**" entry.
2. Enter "**Degree**" as the attribute name, and select "xs:string" as the attribute type.
3. Click the **Enumerations** tab of the Facets entry helper.



4. Click the Append icon of the Enumerations tab and enter "BA", confirm with Enter.
5. Use the same method to add two more items to the enumerations list ("MA" and "Ph.D").

The finished schema should look like this:



element	**Company**	ann:Comment descr
complexType	**AddressType**	ann:
complexType	**US-Address**	ann:
simpleType	**US-State**	ann:
complexType	**UK-Address**	ann:
element	**Person**	ann:

Attributes

Name	Type	Use	Default	Fixed
Manager	xs:boo	required		
Programm	xs:boo	optional		
Degree	xs:strir			

Components

- element
  - **Company**
  - **Person**
- complexType

Details

name	Degree
isRef	□
type	xs:string
default	

Facets

BA
MA
Ph.D

6. Select the menu command **File | Save As**, and save the file as **AddressLast.xsd**.

Please note:
This schema is available as '**AddressLast.xsd**' in the Tutorial folder.

Navigation shortcuts in schema documents

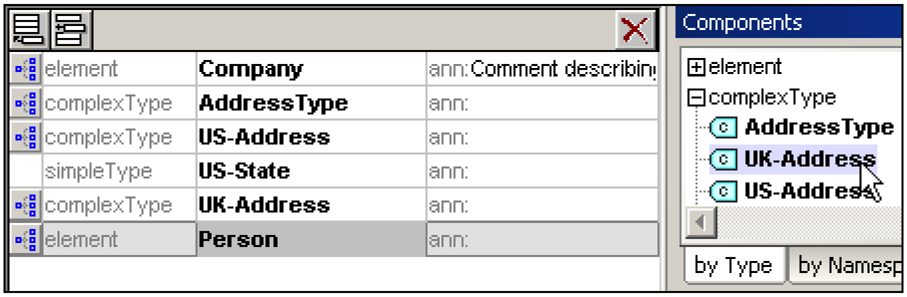
This section is designed to show you how you can navigate the Schema view efficiently.

Displaying the content model of any element:

- Select the **element type** you want to see by clicking the specific Component navigator **tab** e.g. Com(plex).

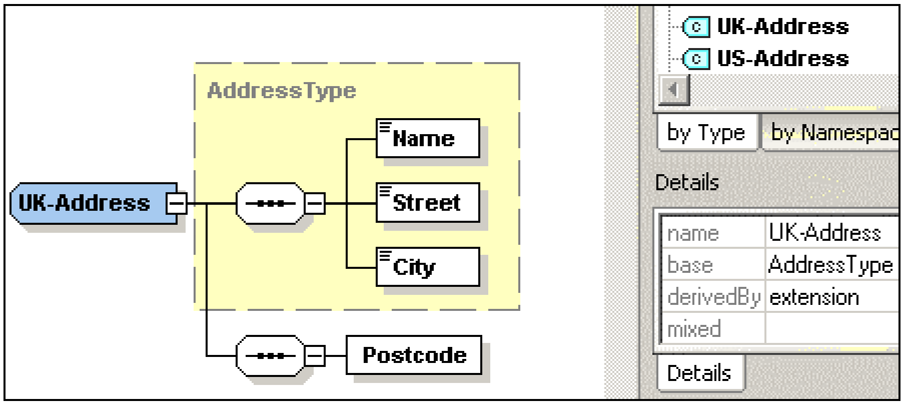
  Elm=global elements, Grp=element group, Com=Complex type, Sim=Simple type, Att=Attribute, AGrp=Attribute group. The Component navigator entries are independent of the content model currently visible in the main window.

- **Double click** the element name in the Com tab e.g. **UK-Address**.



element	**Company**	ann:Comment describin
complexType	**AddressType**	ann:
complexType	**US-Address**	ann:
simpleType	**US-State**	ann:
complexType	**UK-Address**	ann:
element	**Person**	ann:

The content model of the UK-Address element is displayed. The specific settings are shown in the Details tab.

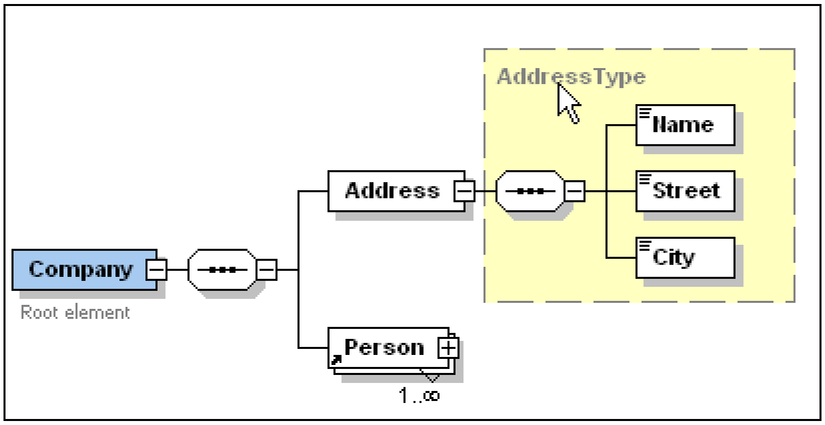


name	UK-Address
base	AddressType
derivedBy	extension
mixed	

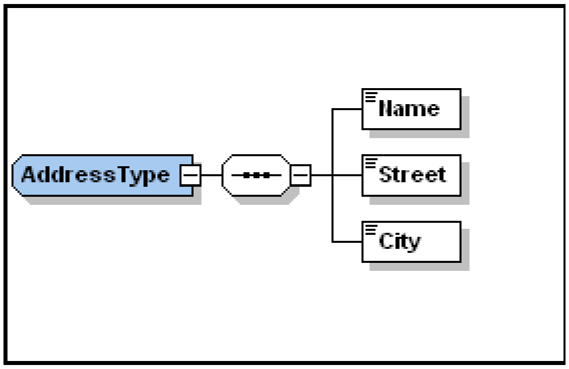
Go to "ElementType" definition:

E.g. While viewing the **Company** content model:

- Double clicking the *AddressType* **text** in the yellow box, takes you to the AddressType definition.



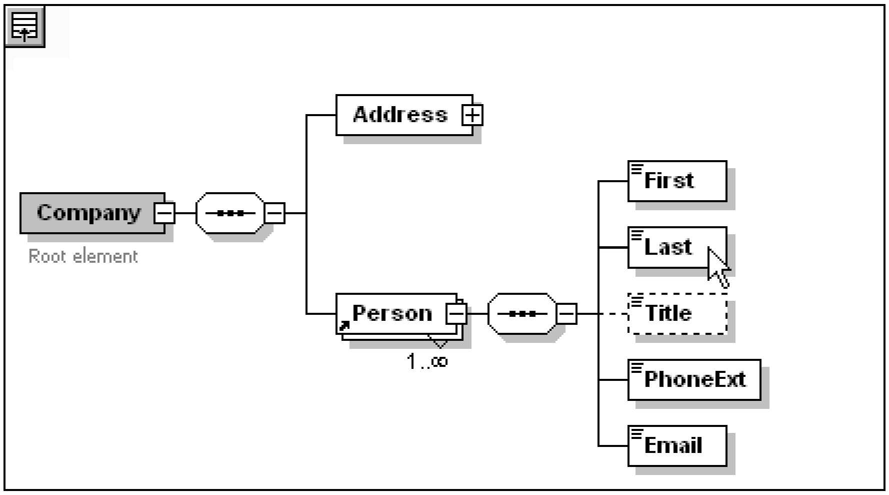
The AddressType definition:



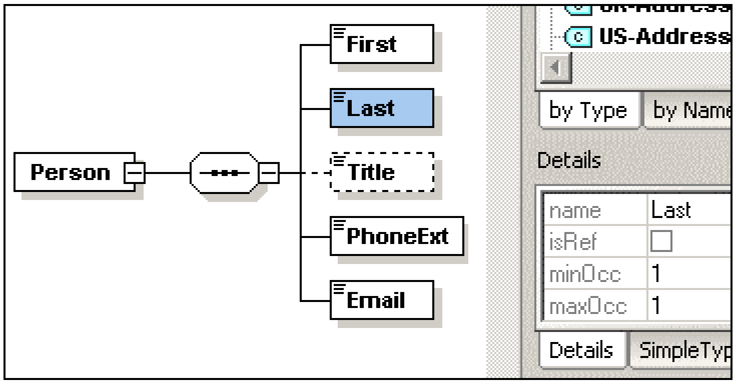
Go to element definition:

E.g. While viewing the **Company** content model:

- Press and hold down the **CTRL** keyboard key, and
- **Double click** on any element definition you want to see (here, the element **Last**).



The element Last, which is a sub-element of the Person element, is displayed. The specific settings are shown in the Details tab.



Details

name	Last
isRef	□
minOcc	1
maxOcc	1

Creating an XML document

Goal of this section:
To create a new XML document and use the various XMLSpy 2005 views and intelligent editing capabilities, to rapidly enter and validate data.

This will be achieved by:

- Creating a new XML document based on the **AddressLast-Home** schema
- Adding elements using **intelligent entry helpers** in Authentic View
- **Validating** the XML document

Functions (and their icons) in this section:

File | New, creates a new type of XML file.

View | Text View, switches to the Text view.

View | Authentic View, switches to Authentic View.

Checks for well-formedness. Hotkey: **F7**.

Validates the XML file against the associated DTD or Schema. Hotkey **F8**.

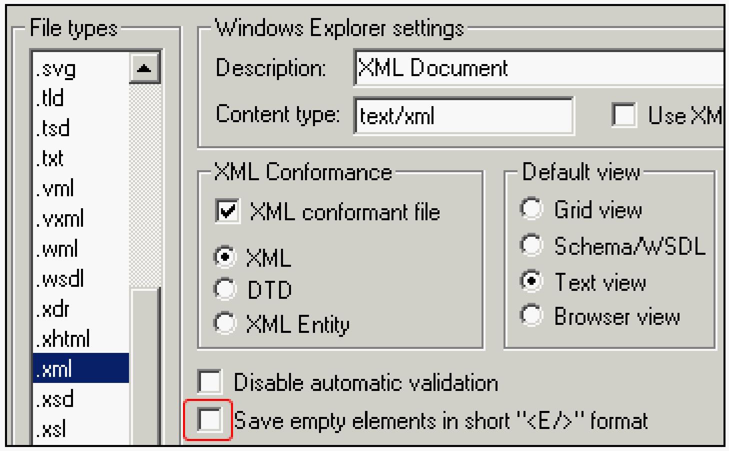
Please note:

> The XML file used in the following section uses a slightly simplified schema file (**AddressLast-home.xsd**) to the one you created in the schema section of this tutorial.

Creating a new XML file

Before we create an XML instance file based on a schema, let's make sure the short format method of creating/saving empty elements is inactive.

1. Select the menu option **Tools | Options**, and then click the **File types** tab.
2. Click the **xml** entry in the list box, and deactivate the "Save empty elements in short "<E/>" format checkbox.



File types

.svg
.tld
.tsd
.txt
.vml
.vxml
.wml
.wsdl
.xdr
.xhtml
.xml
.xsd
.xsl

Windows Explorer settings

Description: XML Document

Content type: text/xml

□ Use XM

XML Conformance

☑ XML conformant file

● XML

○ DTD

○ XML Entity

Default view

○ Grid view

○ Schema/WSDL

● Text view

○ Browser view

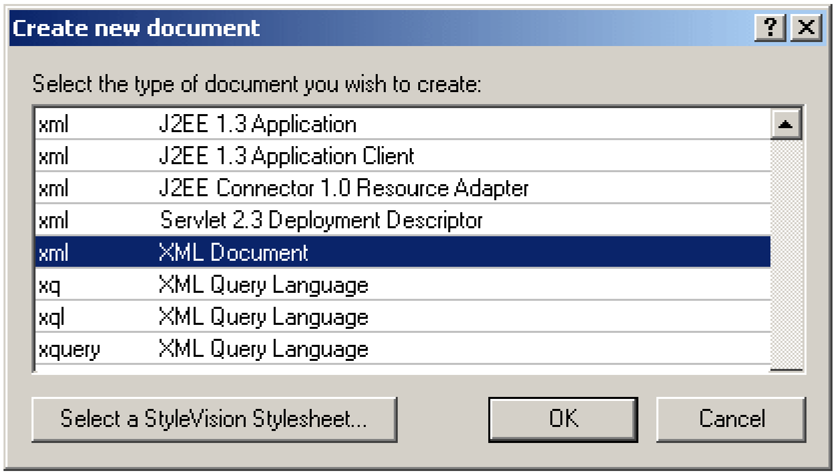
□ Disable automatic validation

□ Save empty elements in short "<E/>" format

3. Click OK to confirm the new settings.

To create a new XML document:

1. Select the menu option **File | New**, and select the **.xml XML Document** entry from the dialog, then confirm with OK.

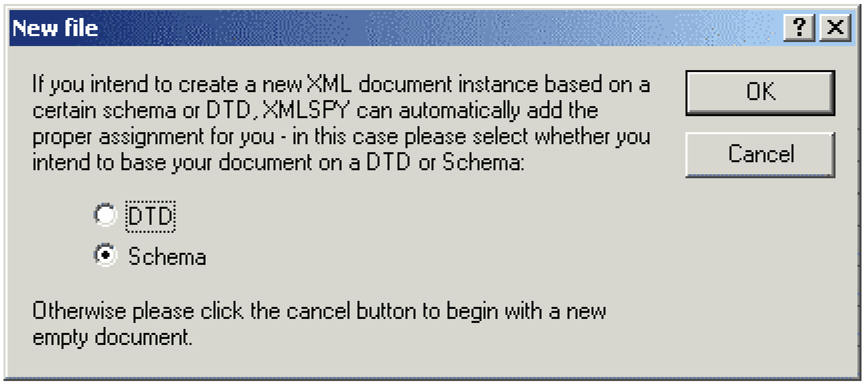


Select the type of document you wish to create:

xml	J2EE 1.3 Application
xml	J2EE 1.3 Application Client
xml	J2EE Connector 1.0 Resource Adapter
xml	Servlet 2.3 Deployment Descriptor
xml	XML Document
xq	XML Query Language
xql	XML Query Language
xquery	XML Query Language

A prompt appears, asking if you want to base the XML document on a DTD or Schema.

2. Click the **Schema** radio button, and confirm with OK.



If you intend to create a new XML document instance based on a certain schema or DTD, XMLSPY can automatically add the proper assignment for you - in this case please select whether you intend to base your document on a DTD or Schema:

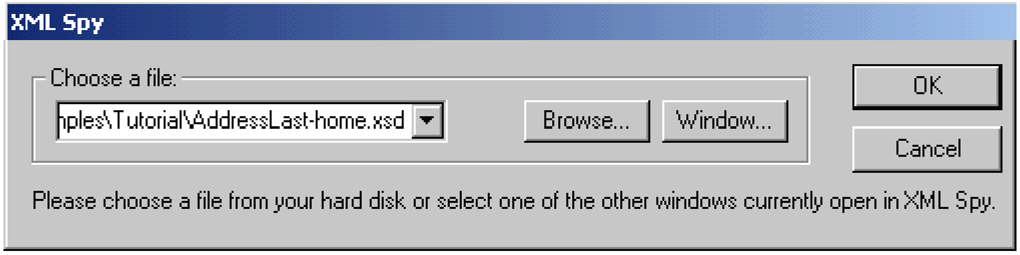
○ DTD

● Schema

Otherwise please click the cancel button to begin with a new empty document.

A further dialog appears, asking you to select the schema file your XML document is to be based on.

3. Use the Browse or Window buttons to find the schema file, in our case the **AddressLast-home** schema, and confirm the selection with OK.



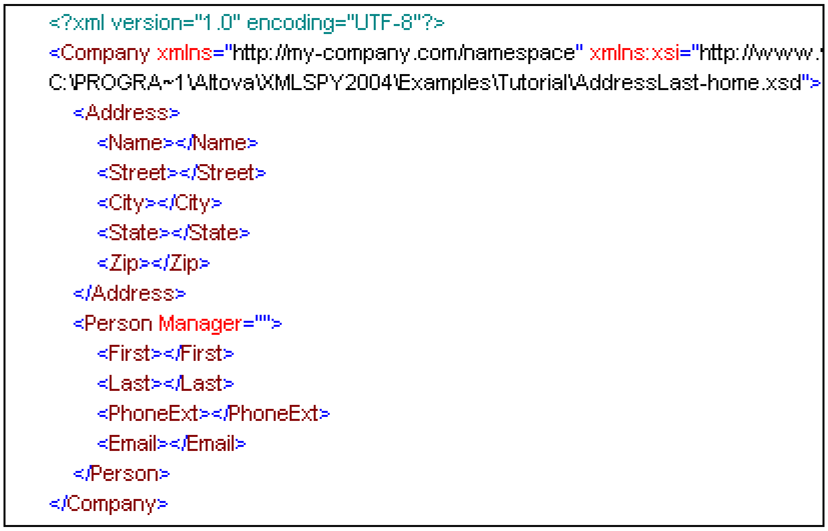
Choose a file: hples\Tutorial\AddressLast-home.xsd

Please choose a file from your hard disk or select one of the other windows currently open in XML Spy.

An XML document containing the main elements defined by the schema, opens in the main window.

Please note:

XMLSpy 2005 tries to find the root element of a schema automatically. The "Wählen Sie das Root Element" dialog box is opened, if it is unclear which is the root element. You can then select the root element manually.



```
<?xml version="1.0" encoding="UTF-8"?>
<Company xmlns="http://my-company.com/namespace" xmlns:xsi="http://www.
C:\PROGRA~1\Altova\XMLSPY2004\Examples\Tutorial\AddressLast-home.xsd">
   <Address>
      <Name></Name>
      <Street></Street>
      <City></City>
      <State></State>
      <Zip></Zip>
   </Address>
   <Person Manager="">
      <First></First>
      <Last></Last>
      <PhoneExt></PhoneExt>
      <Email></Email>
   </Person>
</Company>
```

4. Click on any element to deselect the data.

Editing in Authentic View

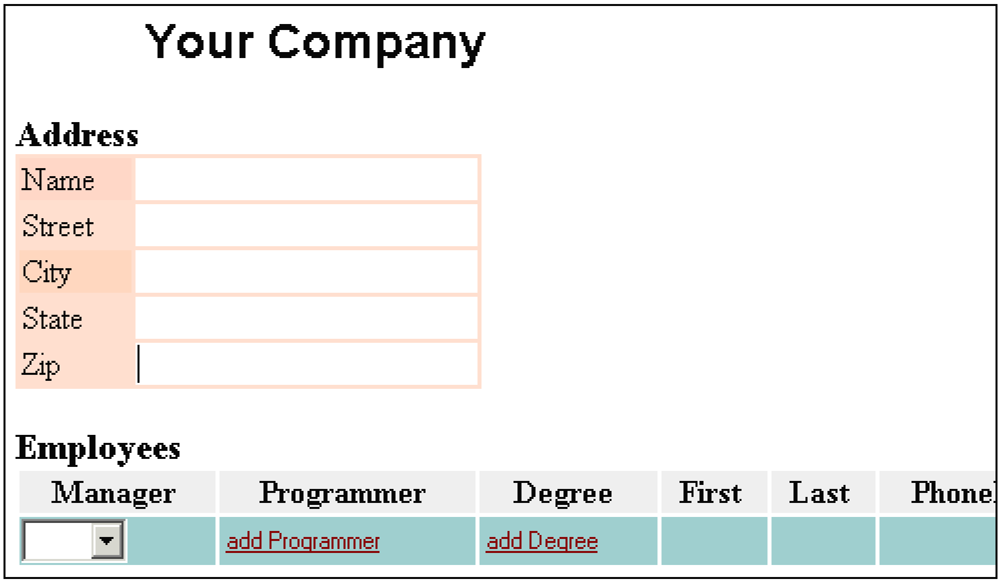
Authentic View enables you to create and edit an XML document that is based on an **Authentic Stylesheet (.sps or SPS file) created in StyleVision 2005**. It is a flexible and easy-to-use interface that features WYSIWYG capabilities and familiar data-entry devices. This enables even users unfamiliar with XML to easily create and edit an XML document.

The Authentic Stylesheet to which an XML document is linked is created in StyleVision by the person who designs the document—and not by you, the user of Authentic View.

Authentic Stylesheets (SPS files) for commonly used schemas are available in the **..\Template\Examples** folder. You can open a new XML document in Authentic View by selecting an Authentic template SPS file in the **File | New** dialog. Alternatively, you can open a new XML document by browsing for the required Authentic Stylesheet. Note that the Authentic Stylesheet (SPS file) is created in StyleVision 2005—not in Authentic View.

Assigning an Authentic Stylesheet to an XML document

1. Select the menu option **Authentic | Assign a StyleVision stylesheet** and hit OK when the prompt appears, to reparse the XML text.
2. Select the **AddressLast-home.sps** file from the ..\Examples\Tutorial folder, and hit OK. The assignment is added to the XML file.
3. Click the **Authentic** tab to switch to the Authentic View.



Your Company

Address

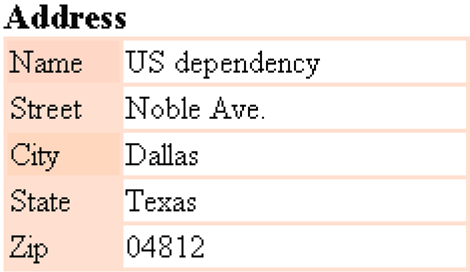
Name	
Street	
City	
State	
Zip	

Employees

Manager	Programmer	Degree	First	Last	Phone
▾	add Programmer	add Degree			

Entering (and deleting) data

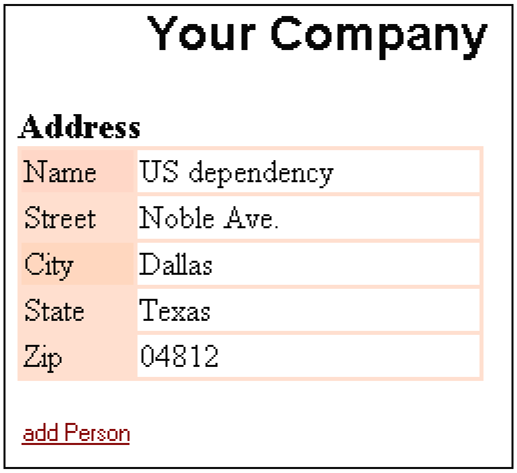
1. Double click in the **Name** value field (or use the arrow keys) and enter "US dependency", then hit the TAB key to move into the next field.
2. Use the same method to enter Street and City names (e.g. Noble Ave. and Dallas etc.), and following on, enter the State and Zip code.



Address

Name	US dependency
Street	Noble Ave.
City	Dallas
State	Texas
Zip	04812

3. Click in the **First** field of the Employees table and select the menu option **Authentic | Delete row** (we will add it again in a few moments in the Text view!).



Your Company

Address

Name	US dependency
Street	Noble Ave.
City	Dallas
State	Texas
Zip	04812

Please note:

The **add Person** placeholder that you now see below the address table is a feature of Authentic View. Clicking the placeholder text would automatically add the Person table in this view.

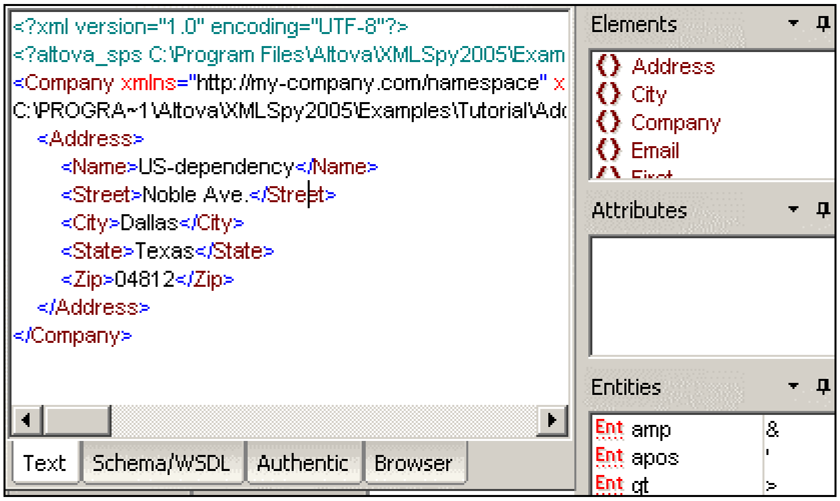
Editing in Text View

XMLSpy 2005 Text view

When it comes down to low-level work, the Text view of XMLSpy 2005 is suitable for editing any type of XML files in textual or source code form, and provides **intelligent editing** capabilities if you are working with an XML document based on a DTD or XML Schema.

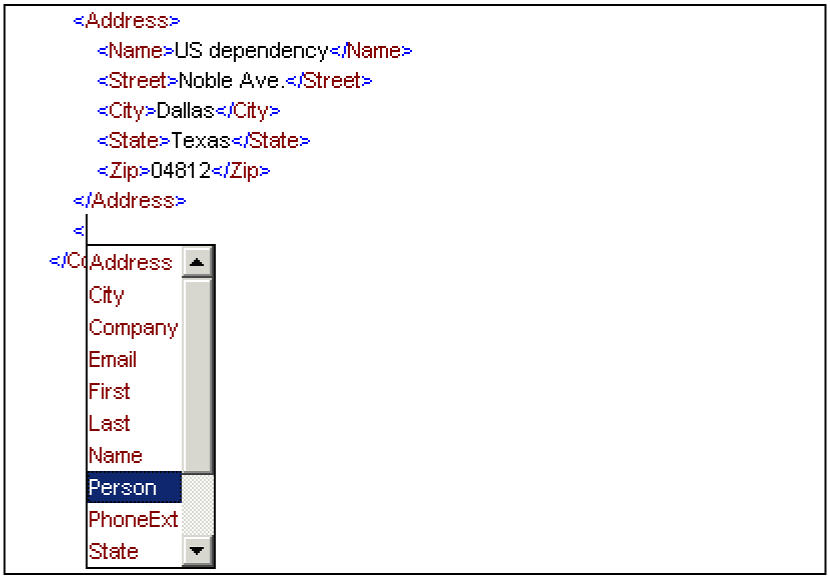
Viewing and entering data in the Text view:

1. Select the menu item **View | Text view**, or click Text tab.
   You now see the XML document in its raw text form (with syntax coloring).



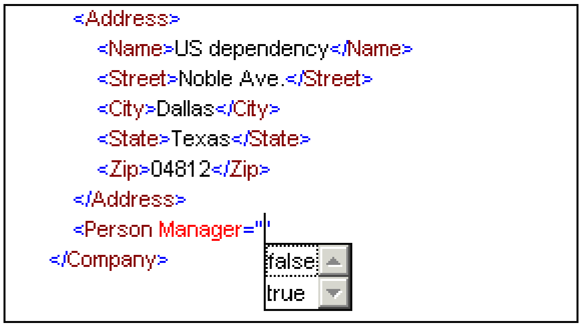
```
<?xml version="1.0" encoding="UTF-8"?>
<?altova_sps C:\Program Files\Altova\XMLSpy2005\Exam
<Company xmlns="http://my-company.com/namespace" x
C:\PROGRA~1\Altova\XMLSpy2005\Examples\Tutorial\Adc
  <Address>
    <Name>US-dependency</Name>
    <Street>Noble Ave.</Street>
    <City>Dallas</City>
    <State>Texas</State>
    <Zip>04812</Zip>
  </Address>
</Company>
```

2. Place the text cursor after the </Address> | **end tag**, and hit Enter to add a new line.
3. Enter the "less than" angle bracket **<** at this position.



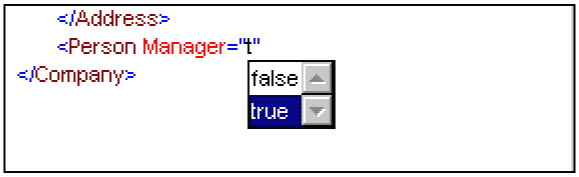
```
  <Address>
    <Name>US dependency</Name>
    <Street>Noble Ave.</Street>
    <City>Dallas</City>
    <State>Texas</State>
    <Zip>04812</Zip>
  </Address>
  <
</C
```

4. A drop-down list appears; select the **Person** entry.
   The element name "Person" as well as the attribute "Manager", are inserted. The attribute value drop-down list is also automatically opened.



```
  <Address>
    <Name>US dependency</Name>
    <Street>Noble Ave.</Street>
    <City>Dallas</City>
    <State>Texas</State>
    <Zip>04812</Zip>
  </Address>
  <Person Manager=""
</Company>
```

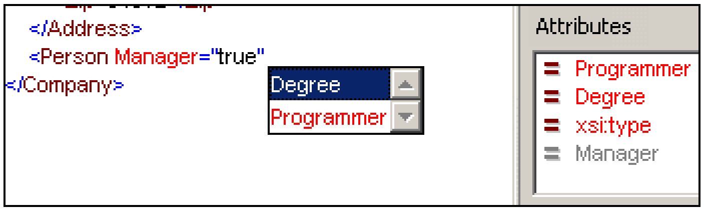
5. Enter the letter "**t**" and hit Enter.



```
    </Address>
    <Person Manager="t"
</Company>
```

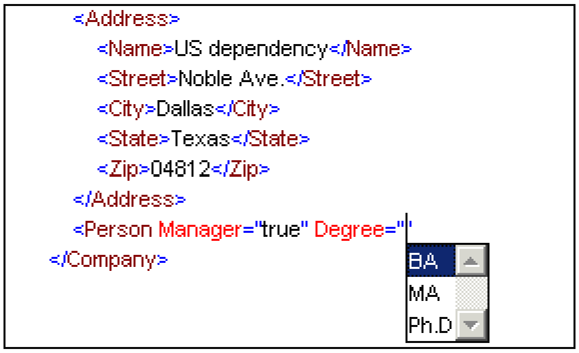
This selects the attribute "true" and inserts it at the cursor position.

6. Move the cursor to the end of the line (End key), and hit the spacebar.
   This opens the drop-down list again. There are now fewer entries available in the list; "Manager" is grayed out in the Attribute entry helper.



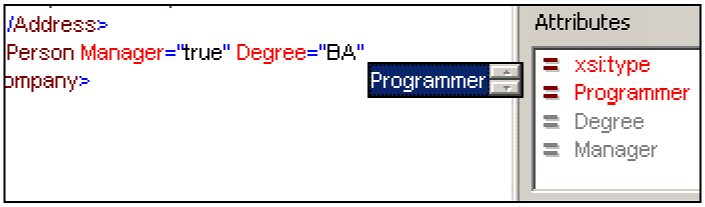
```
  </Address>
  <Person Manager="true"
</Company>
```

7. Select "Degree" with the Down arrow key, and hit Enter.
   This opens a further drop-down list from which you can select one of the predefined enumerations (BA, MA or Ph.D).



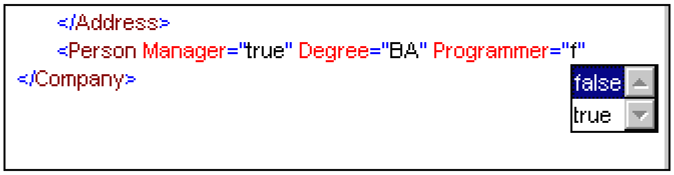
```
    <Address>
      <Name>US dependency</Name>
      <Street>Noble Ave.</Street>
      <City>Dallas</City>
      <State>Texas</State>
      <Zip>04812</Zip>
    </Address>
    <Person Manager="true" Degree="
  </Company>
```

8. Select "BA" with the Down arrow key (confirm with OK), move the cursor to the end of the line (End key), and hit the spacebar. Manager and Degree are now grayed out in the Attribute entry helper.



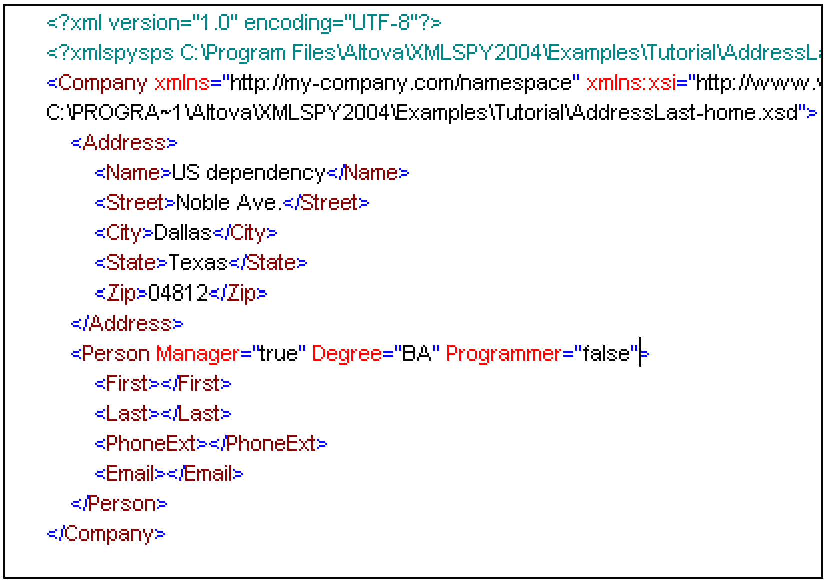
```
/Address>
Person Manager="true" Degree="BA"
mpany>
```

9. Select "Programmer" with the Down arrow key, and hit Enter.



```
    </Address>
    <Person Manager="true" Degree="BA" Programmer="f"
</Company>
```

10. Enter a "**f**" character and hit Enter.
11. Move the cursor to the end of the line (End key), and enter the "greater than" angle bracket **>**.



```
<?xml version="1.0" encoding="UTF-8"?>
<?xmlspysps C:\Program Files\Altova\XMLSPY2004\Examples\Tutorial\AddressL
<Company xmlns="http://my-company.com/namespace" xmlns:xsi="http://www.
C:\PROGRA~1\Altova\XMLSPY2004\Examples\Tutorial\AddressLast-home.xsd">
  <Address>
    <Name>US dependency</Name>
    <Street>Noble Ave.</Street>
    <City>Dallas</City>
    <State>Texas</State>
    <Zip>04812</Zip>
  </Address>
  <Person Manager="true" Degree="BA" Programmer="false">
    <First></First>
    <Last></Last>
    <PhoneExt></PhoneExt>
    <Email></Email>
  </Person>
</Company>
```

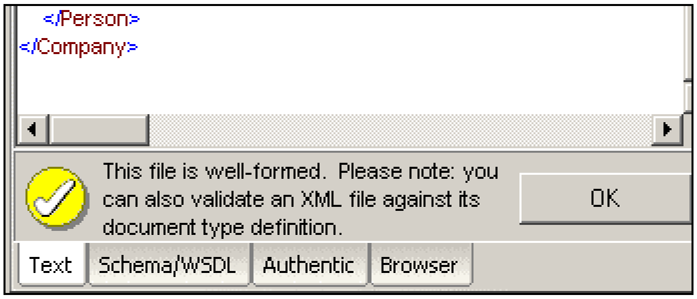
XMLSpy 2005 automatically inserts all the Person element tags. Each element is supplied with start and end tags.

Validating and entering data

At this point let's check if the document is well-formed and valid, there might still be work to do.

To check for well-formedness:

1. Select the menu option **XML | Check well-formedness** or hit the **F7** key.
   A message appears at the bottom of the main window declaring that the document is well formed. Click OK to confirm and close the message.



```
  </Person>
</Company>
```

This file is well-formed. Please note: you can also validate an XML file against its document type definition.

Being well-formed, means that the XML document **syntax** is correct (i.e. there is a root element, each start tag has a corresponding end tag, all elements are nested correctly etc.).

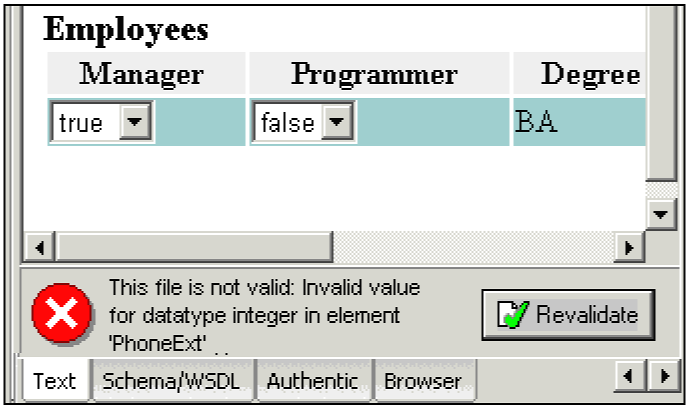
This check does not check against a schema file (or any other external file). Element sequence or element content are not checked either. Well-formedness can only be checked in the Text view.

To check for validity:

The **validity** of an XML document can be checked both in the Text and Authentic View.

1. Click the **Authentic** tab to switch into Authentic View.
2. Select the menu option **XML | Validate** or hit the **F8** key.
   An **error message** appears: "This file is not valid: Invalid value for datatype integer in element "PhoneExt".

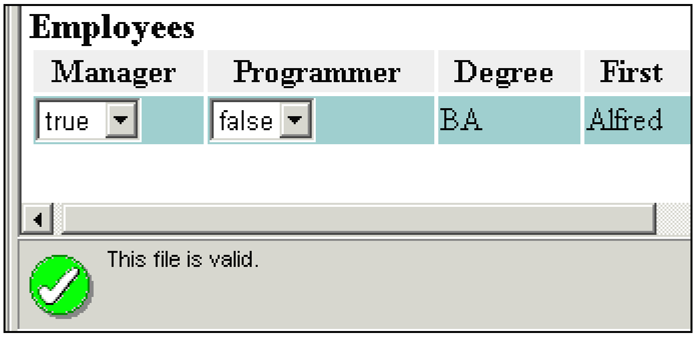
The error message describes in detail what is currently wrong with our XML document. In this case the datatype is flagged as incorrect because it is empty, we have not yet entered any data in this field.



Employees		
Manager	Programmer	Degree
true	false	BA

This file is not valid: Invalid value for datatype integer in element 'PhoneExt' . .

3. Fill in the rest of the table fields (e.g. Alfred, Aldrich, PhoneExt=33).
4. Click the **Revalidate** button to check if the document is now valid.
   The "This file is valid" message appears. The XML document is now valid against its schema. Click OK to confirm and close the message.



Employees			
Manager	Programmer	Degree	First
true	false	BA	Alfred

This file is valid.

Being valid, means that the XML document adheres to the assigned schema i.e. the elements and the sequence they appear in is correct, as well as the element "contents" and their attributes.

5. Select the menu option **File | Save As...** and name the XML document (e.g. **CompanyLast-Home.xml**)

Please note:

An XML document does not have to be valid in order to save it. Saving an invalid document causes a prompt to appear which then allows you to select "**Save anyway**", the document is then saved in its current state.

Authentic View provides **real-time** content **validation** against the referenced schema. Whenever invalid data is entered, it is automatically highlighted in red. Hitting F8 provides more information on why the data is invalid.

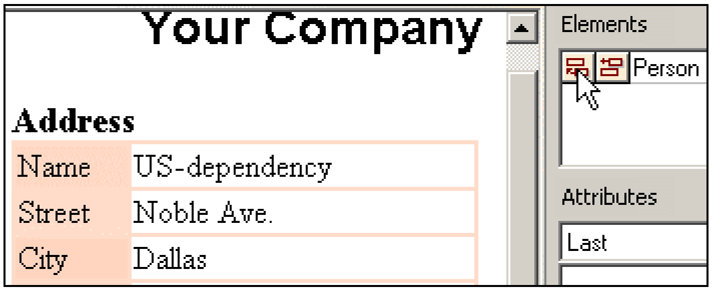
Positioning the text cursor over the Degree field for a few seconds, displays a tooltip describing the type of data that must be entered here. The tooltip text is defined in StyleVision 2005.

Manipulating data with Entry Helpers

At this point we want to enter more employees to the XML document.

Inserting elements and attributes (intelligent entry help):

1. Click into one of the table fields.



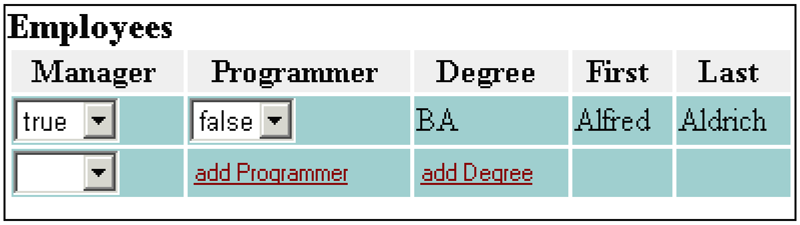
Your Company

Address

Name	US-dependency
Street	Noble Ave.
City	Dallas

You will notice that the Person element is now visible in the Elements entry helper. The entries in the Element and Attributes entry helpers are dependent on where you click in the Authentic document.

2. Click the append icon to append a person row to the employees table.

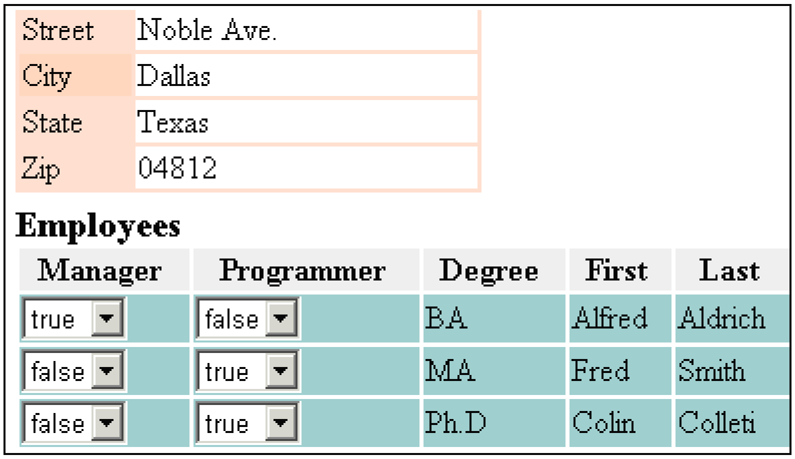


Employees

Manager	Programmer	Degree	First	Last
true	false	BA	Alfred	Aldrich
	add Programmer	add Degree		

The Manager combo box is visible because the attribute was defined as "required" in the associated schema. The **add...** (element/attribute) placeholders signal that the element or attribute is defined as optional. Clicking the placeholder enables you to enter data at that point.

3. Fill in the rest of the table data.



Street	Noble Ave.
City	Dallas
State	Texas
Zip	04812

Employees

Manager	Programmer	Degree	First	Last
true	false	BA	Alfred	Aldrich
false	true	MA	Fred	Smith
false	true	Ph.D	Colin	Colleti

The XML document shown above, is available as 'CompanyLast-Home.xml' in the ..Examples\Tutorial folder.

XSL Transformation

Objective

To generate an HTML file from the XML file using an XSL stylesheet to transform the XML file. You should note that a "transformation" does not change the XML file into anything else; instead a new output file is generated. The word "transformation" is a convention.

Method

The method used to carry out the transformation is as follows:

- Assign a predefined XSL file, `Company.xsl`, file to the XML document.
- Execute the transformation within the XMLSpy 2005 interface using one of the two built-in Altova XSLT engines. (*See note below.*)

The following XMLSpy 2005 commands are used in this section:

XSL/XQuery | Assign XSL, which assigns an XSL file to the active XML document.

XSL/XQuery | Go to XSL, opens the XSL file referenced by the active XML document.

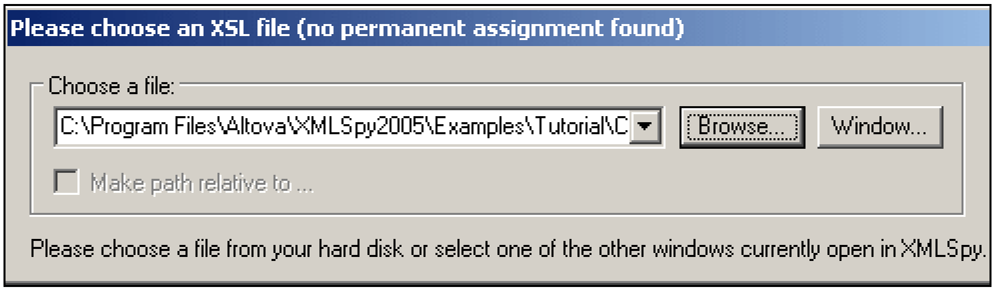
XSL/XQuery | XSL Transformation (F10), or the toolbar icon , transforms the active XML document using the XSL stylesheet assigned to the XML file. If an XSL file has not been assigned then you will be prompted for one when you select this command.

Note: XMLSpy 2005 has two built-in XSLT engines, the Altova XSLT 1.0 Engine and Altova XSLT 2.0 Engine. The Altova XSLT 1.0 Engine is used to process XSLT 1.0 stylesheets. The Altova XSLT 2.0 Engine is used to process XSLT 2.0 stylesheets. The correct engine is automatically selected by XMLSpy 2005 on the basis of the version attribute in the `xsl:stylesheet` or `xsl:transform` element. In this tutorial transformation, we use XSLT 1.0 stylesheets. The Altova XSLT 1.0 Engine will automatically be selected for transformations with these stylesheets when the **XSL Transformation** command is invoked.

Assigning an XSL file

To assign an XSL file to the `CompanyLast.xml` file, do the following:

1. Click the `CompanyLast.xml` tab in the main window so that `CompanyLast.xml` becomes the active document.
2. Select the menu command **XSL/XQuery | Assign XSL**.
3. Click the Browse button, and select the `Company.xsl` file from the Tutorial folder. In the dialog, you can check the option Make Path Relative to CompanyLast.xml if you wish to make the path to the XSL file (in the XML document) relative.



Please choose an XSL file (no permanent assignment found)

Choose a file:

C:\Program Files\Altova\XMLSpy2005\Examples\Tutorial\C

□ Make path relative to ...

Please choose a file from your hard disk or select one of the other windows currently open in XMLSpy.

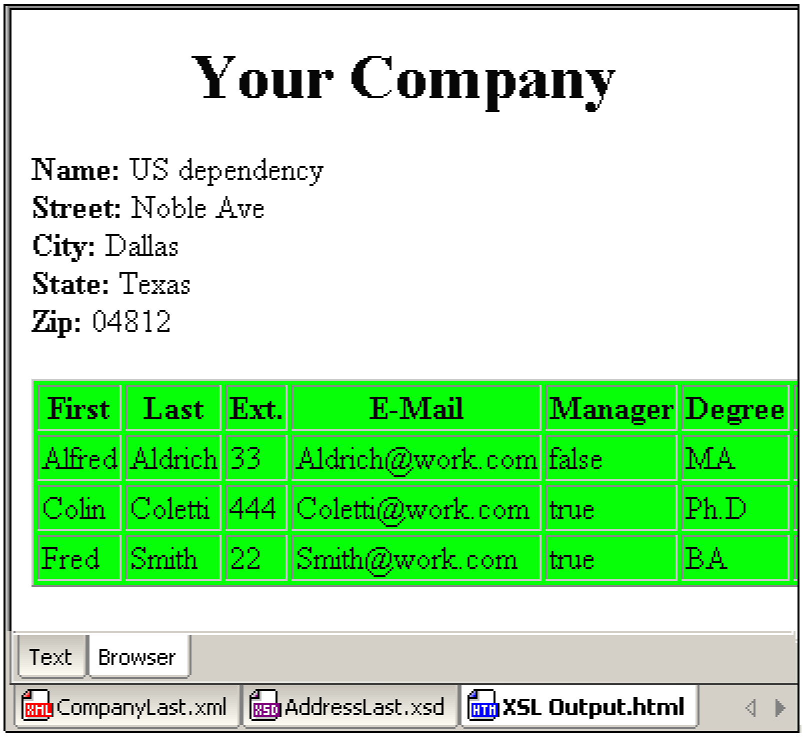
4. Click the OK button to assign the XSL file to the XML document.

   An `XML-stylesheet` processing instruction is inserted in the XML document that references the XSL file. If you have checked the Make Path Relative to CompanyLast.xml check box, then the path is relative; otherwise absolute (as in the screenshot above).

Transforming the XML file

To transform the XML document using the XSL file you have assigned to it, do the following:

1. Ensure that the XML file is the active document.
2. Select the menu option **XSL/XQuery | XSL Transformation (F10)** or click the icon. This starts the transformation using the XSL stylesheet referenced in the XML document. (Since the `Company.xsl` file is an XSLT 1.0 document, the built-in Altova XSLT 1.0 Engine is automatically selected for the transformation.) The output document is displayed in Browser View; it has the name `XSL Output.html`. It shows the Company data in one block down the left, and the Person data in tabular form below.



Your Company

Name: US dependency
Street: Noble Ave
City: Dallas
State: Texas
Zip: 04812

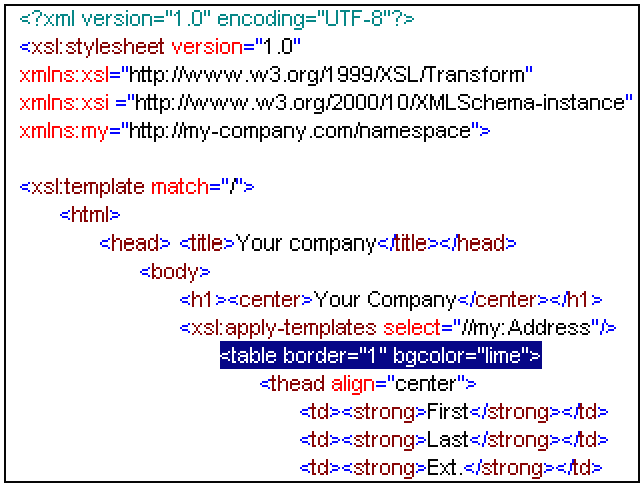
First	Last	Ext.	E-Mail	Manager	Degree
Alfred	Aldrich	33	Aldrich@work.com	false	MA
Colin	Coletti	444	Coletti@work.com	true	Ph.D
Fred	Smith	22	Smith@work.com	true	BA

Note: Should you only see a table header and no table data in the output file, please make sure that you have defined the target namespace for your schema as detailed in Defining your own namespace at the beginning of the tutorial. The namespace must be **identical** in all three files (Schema, XML, and XSL).

Modifying the XSL file

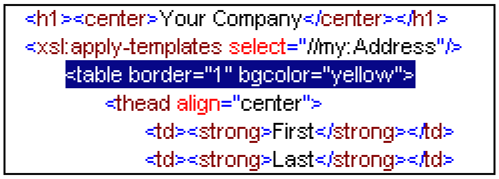
You can change the output by modifying the XSL document. For example, let's change the background-color of the table in the HTML output from lime to yellow. This is how you would do it:

1. Click **File | Open**, and browse for the `Company.xsl` file, which is in the `Examples/Tutorials` folder. This command opens the `Company.xsl` file (*screenshot below*).



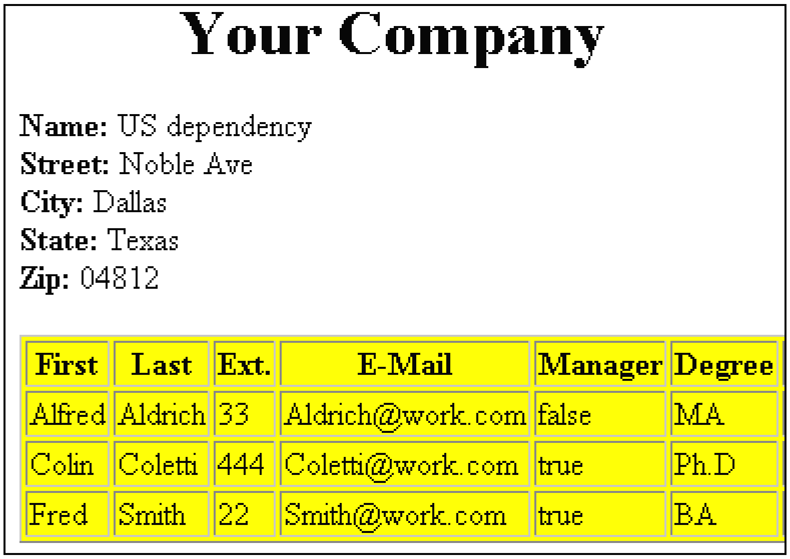
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi ="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:my="http://my-company.com/namespace">

<xsl:template match="/">
    <html>
        <head> <title>Your company</title></head>
            <body>
                <h1><center>Your Company</center></h1>
                <xsl:apply-templates select="//my:Address"/>
                    <table border="1" bgcolor="lime">
                        <thead align="center">
                            <td><strong>First</strong></td>
                            <td><strong>Last</strong></td>
                            <td><strong>Ext.</strong></td>
```

2. Find the line `<table border="1" bgcolor="lime">`, and change the entry `bgcolor="`**`lime`**`"` to `bgcolor="`**`yellow`**`"`.



```
<h1><center>Your Company</center></h1>
<xsl:apply-templates select="//my:Address"/>
    <table border="1" bgcolor="yellow">
        <thead align="center">
            <td><strong>First</strong></td>
            <td><strong>Last</strong></td>
```

3. Select the menu option **File | Save** to save the changes made to the XSL file.
4. Click the `CompanyLast.xml` tab to make the XML file active, and select **XSL/XQuery | XSL Transformation,** or hit the **F10** key. A new `XSL Output.html` file appears in the XMLSpy 2005 GUI in Browser View. The background-color of the table is yellow.



Your Company

Name: US dependency
Street: Noble Ave
City: Dallas
State: Texas
Zip: 04812

First	Last	Ext.	E-Mail	Manager	Degree
Alfred	Aldrich	33	Aldrich@work.com	false	MA
Colin	Coletti	444	Coletti@work.com	true	Ph.D
Fred	Smith	22	Smith@work.com	true	BA

5. Select the menu option **File | Save**, and save the document as `Company.html`.

3.2 Authentic View Tutorial

In Authentic View, you will open an existing XML file that is linked to a StyleVision Power Stylesheet. You then modify the file using the various Authentic View features. The tutorial consists of three broad parts:

- Opening an existing XML file in Authentic View
- Editing data (adding new document components as well as content); this section forms the bulk of the tutorial
- Printing out the document

Remember that this tutorial is intended to get you started, and has intentionally been kept simple. You will find additional reference material and feature descriptions in the Authentic View interface sections.

Tutorial requirements
All **the files** you need for the tutorial are in the `Examples` folder of your Altova application folder. These files are :

- `NanonullOrg.xml` (the XML document you will open)
- `NanonullOrg.sps` (the StyleVision Power Stylesheet to which the XML document is linked)
- `NanonullOrg.xsd` (the XML Schema on which the XML document and StyleVision Power Stylesheet are based, and to which they are linked)
- `nanonull.gif` and `Altova_right_300.gif` (two image files used in the tutorial)

Note: At some points in the tutorial, we ask you to look at the XML text of the XML document (as opposed to the Authentic View of the document). If the Altova product edition you are using does not include a Text View (as in the case of the free Authentic 2005 Desktop Edition and Authentic 2005 Browser Edition), then use a plain **text editor** like Wordpad or Notepad to view the text of the XML document.

Caution: We recommend that you use a copy of `NanonullOrg.xml` for the tutorial, so that you can always retrieve the original should the need arise.

3.2.1 Opening an XML document in Authentic View

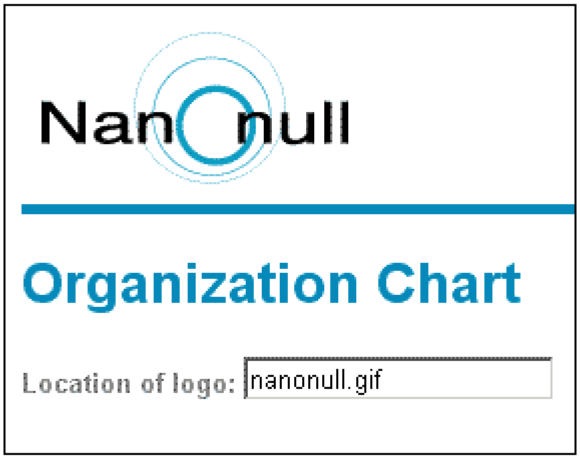
The file `NanonullOrg.xml` is in the `Examples` folder in the folder where you have installed the application. Typically, the path to the file would be:
`c:\Program\Files\Altova\XMLSpy 2005\Examples\NanonullOrg.xml`

You can open `NanonullOrg.xml` in one of two ways:

- Click **File | Open** in your Altova product, then browse for `NanonullOrg.xml` in the dialog that appears, and click Open.
- Use Windows Explorer to locate the file, right-click, and select your Altova product.

The file `NanonullOrg.xml` opens directly in Authentic View. This is because:

1. The file already has a StyleVision Power Stylesheet assigned to it.
2. In the Options dialog (**Tools | Options**), in the View tab, the option to open XML files in Authentic View if an SPS file is assigned has been checked. (Otherwise the file would open in Text View.)



Organization Chart

Location of logo: nanonull.gif

Remember: It is the StyleVision Power Stylesheet that defines and controls how an XML document is displayed in Authentic View. Without a StyleVision Power Stylesheet, there can be no Authentic View of the document.

Note: Alternatively, you could open an XML template in Authentic View by selecting a StyleVision Power Stylesheet. To open such a template, do the following:

1. Select **File | New**, and, in the Create a New Document dialog, select XML as the new file type to create.
2. Click **Select a STYLEVISION Stylesheet**, and browse for the desired StyleVision Power Stylesheet.

If a Template XML File has been assigned to the StyleVision Power Stylesheet, then the data in the Template XML File is used as the starting data of the template that has been created in Authentic View.

3.2.2 Entering data in Authentic View

Entering data in Authentic View can be as simple as the associated StyleVision Power Stylesheet makes it.

In the **simplest user scenario**, you will enter content as free-flowing text or into data-input fields, or you will make a selection from a list of user options. In short, you focus on entering content; user-side structural modification (adding elements, tables, etc) and document formatting is kept to a minimum. These restrictions help ensure the validity of the document and the accuracy of data. They also keep you focused on the content.

In most cases, however, you will be given the **option of adding a few elements**. These additions can be implicit, as when you press Enter to add a new paragraph element or click an icon to mark text bold. Or they can be explicit, as when you append an element via the Elements Entry Helper. In the latter kind of scenario, you would require a working knowledge of how the document is structured.

At the other extreme, users may be given complete freedom to structure the document. To do this, however, you would require a good knowledge of the schema on which the document is based.

In Authentic View, you can enter or edit the following types of data and data structures:

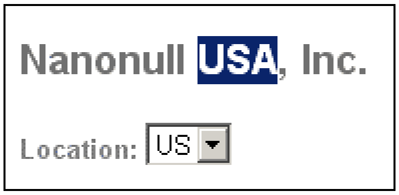
- Element content (can be entered as a text entry or via a data-entry device)
- Attribute values (can be entered as a text entry or via a data-entry device, or as a value in the Attributes Entry Helper)
- Entities (can be inserted via the Entities Entry Helper)
- Elements (can be added, changed to other elements, and deleted)
- XML tables (can be inserted, and its structure, formatting, and content specified)

This tutorial shows you how to manipulate elements and enter content in Authentic View. XML tables are discussed in detail in the Using tables in Authentic View section.

3.2.3 Adding document content and elements

Adding text content
You can enter element content and attribute values directly as text. To insert content, place the cursor at the location where you want to insert the text, and type. You can also edit such content by highlighting the text, and typing in the replacement text or deleting the highlighted text.

To change the name of the company from Nanonull, Inc to Nanonull USA, Inc, place the cursor after Nanonull, and type in the text to be inserted.



Nanonull USA, Inc.

Location: US

If text is editable, you will be able to place your cursor in it and highlight it, otherwise you will not be able to. Try changing any of the field names, "Street", "City", "State/Zip" (in the address block). You are not able to place the cursor in this text because such text is not XML content; it is derived from the StyleVision Power Stylesheet.

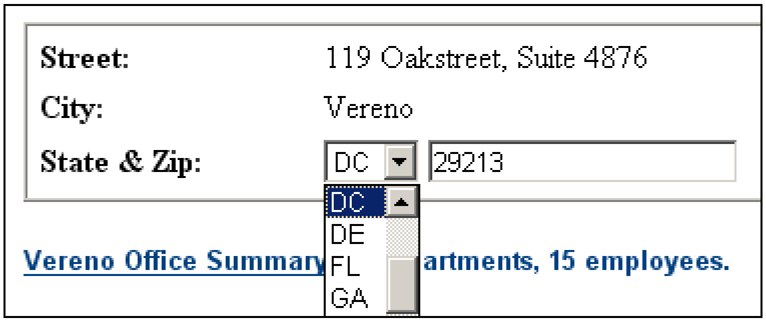
Note: In Hide markup mode, an empty element can easily be overlooked. To make sure that you are not overlooking an empty element, switch to Show large markup or Show small markup mode.

Adding content via a data-entry device
In the content editing you have learned above, content is added by directly typing in text as element content. There is one other way that **element content** (or attribute values) can be entered in Authentic View: via data-entry devices.

Given below is a list of data-entry devices in Authentic View, together with an explanation of how data is entered in the XML file for each device.

Data-Entry Device	Data in XML File
Input Field (Text Box)	Text entered by user
Multiline Input Field	Text entered by user
Combo box	User selection mapped to value
Check box	User selection mapped to value
Radio button	User selection mapped to value
Button	User selection mapped to value

In the static table containing the address fields (shown below), there are two data-entry devices: an input field for the Zip field and a combo-box for the State field. The values that you enter in the text fields are entered directly as the XML content of the respective elements. For other data-entry devices, your selection is mapped to a value.



Street: 119 Oakstreet, Suite 4876
City: Vereno
State & Zip: DC 29213

Vereno Office Summary artments, 15 employees.

For the Authentic View shown above, here is the corresponding XML text:

```
<Address>
  <ipo:street>119 Oakstreet, Suite 4876</ipo:street>
  <ipo:city>Vereno</ipo:city>
  <ipo:state>DC</ipo:state>
  <ipo:zip>29213</ipo:zip>
</Address>
```

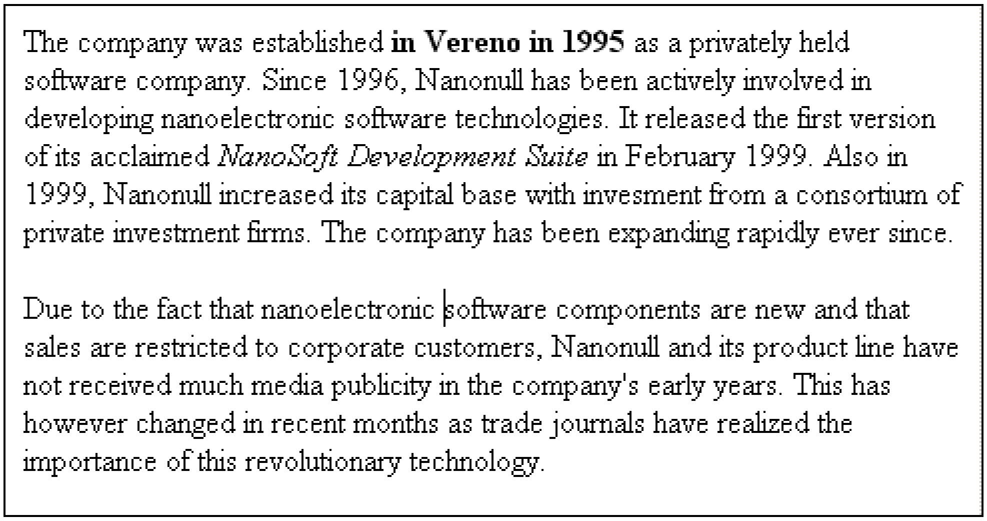
Notice that the combo-box selection "DC" is mapped to a value of "DC". The value of the Zip field is entered directly as content of the `ipo:zip` element.

Adding elements = inserting before or appending

You can add an element by **inserting it before** or **appending it** to the **current element** (the element in which the cursor is).

Add another paragraph to the description of the company. This involves appending the paragraph element (`para`, in this case) and entering content for the element. Do this as follows:

1. Place the cursor in the last paragraph of the description text.
2. Click the (Insert Before Element) icon that is located to the left of the `para` element in the Elements Entry Helper. This **inserts** a `para` **before** the current `para`—which is not where you want the new `para` element.
3. Click the (Append Element) icon that is located to the left of the `para` element in the Elements Entry Helper. This **appends** a `para` element, which is what we want.
4. Type in the content of `para` at the blinking cursor.



The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with invesment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Due to the fact that nanoelectronic software components are new and that sales are restricted to corporate customers, Nanonull and its product line have not received much media publicity in the company's early years. This has however changed in recent months as trade journals have realized the importance of this revolutionary technology.

Alternatively, to append a `para` (or any paragraph-type) element, press **Enter** at the point where you want to append the new element.

Note: You can also use the Enter key to insert and append list items in numbered lists and itemized (bulletted) lists.

Adding elements in the document
It is important to remember that **only same- or higher-level elements** can be inserted before or appended after the current element.

Same-level elements are siblings. Siblings of a paragraph element would be other paragraph elements, but could also be lists, a table, an image, etc. Siblings could occur before or after an element.

Higher-level elements are ancestor elements and siblings of ancestors. For a paragraph element, ancestor elements could be a section, chapter, article, etc. A paragraph in a valid XML would already have ancestors. Therefore, adding a higher-level element in Authentic View, creates it as a sibling of the relevant ancestor. For example, if a section element is appended to a paragraph, it is created as a sibling of the section that contains the current paragraph element, and it is created as the last sibling section.

At any given location in the document, the elements you can insert before or append are shown in the Elements Entry Helper with the icons to their left.

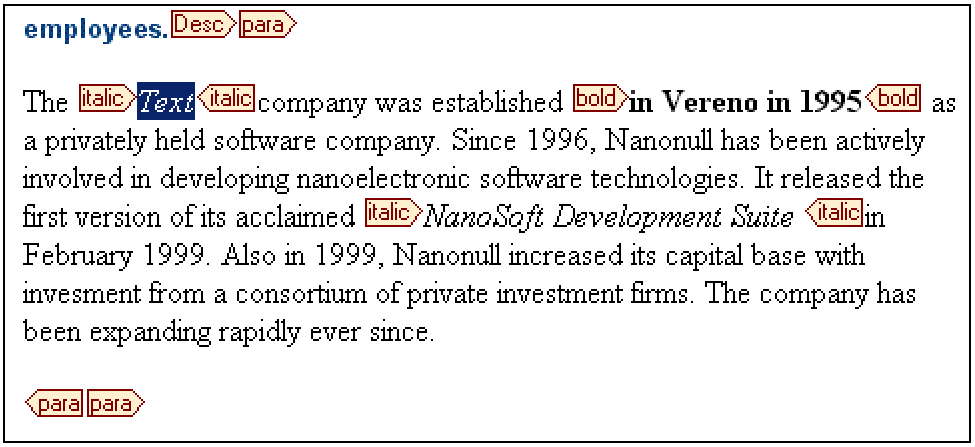
3.2.4 Inserting an element

You can **insert** an element **as a child** of the **current element** (the element in which the cursor is).

In `NanonullOrg.xml`, the `para` element can contain the elements `italic` and `bold`. Now experiment with inserting these elements, as follows:

1. Place the cursor at a location in the `para` element where you want to insert the child element (check the status bar at bottom left for your location in the document). Notice that in the Elements Entry Helper, the `italic` and `bold` elements are listed with (Insert Element) icons next to the listing.
2. Click the icon next to the `italic` element. This inserts the `italic` element with

a placeholder, "Text", which is highlighted. Switch to Show large markup mode to see the element tags.



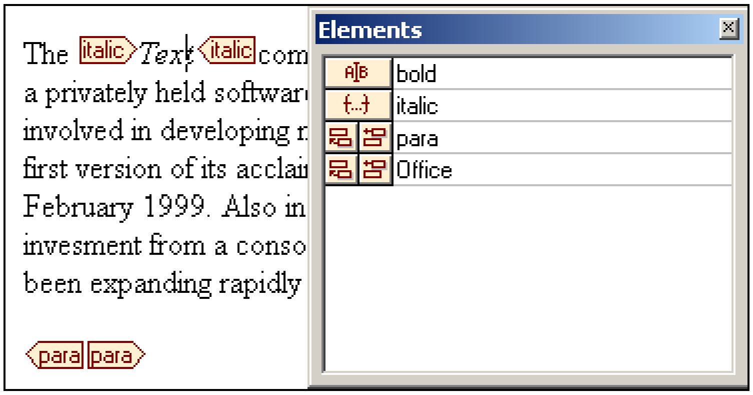
employees.

The *Text* company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with invesment from a consortium of private investment firms. The company has been expanding rapidly ever since.

3. Type in the content of the element.

When the cursor is within an element, any element which can be inserted as a child is shown in the Elements Entry Helper with the Insert Element icon next to it.

Now place the cursor inside the `italic` element, and look at the Elements Entry Helper.



The *Text* com
a privately held softwar
involved in developing n
first version of its acclai
February 1999. Also in
invesment from a conso
been expanding rapidly

The Elements Entry Helper shows that the `bold` element can be inserted as a child of `italic`. Click the Insert Element icon for the `bold` element, and see what happens.

3.2.5 Applying and clearing elements

Applying (replacing) an element

The Apply Element icon AB allows you to replace the selected element with another element allowed at that location.

To apply the `bold` element to the `italic` element you have just created, do the following:

1. Switch to Show large markup mode. It is easier to select an entire element in this mode.
2. Select the `italic` element by clicking either its start or end tag. The `bold` element in the Elements Entry Helper is displayed with the AB (Apply Element) icon next to it.
3. Click the icon. The `italic` element in the main window changes to a `bold` element.

When the entire element is selected, elements which can be applied to the selected element are shown in the Elements Entry Helper with the AB (Apply Element) icon next to it.

Note: It is important to select the entire element, and not just its contents. Selecting the content activates the Clear Element context of the selected element.

Clearing an element
When you clear an element, you remove its markup without modifying its content.

To clear the `italic` element, do the following:

1. Switch to Show large markup mode.
2. Place the cursor as an insertion point within the `italic` element. The `italic` element is shown in the Elements Entry Helper with the {...} icon next to it.
3. Click the {...} icon. In the main window, the `italic` element markup is removed, and the italic formatting is removed.

You can also select a text fragment within the `italic` element (instead of placing the cursor as an insertion point), and clear the `italic` element. The only difference is that the `italic` element in the Elements Entry Helper will be shown with a second variant of the Clear Element icon next to it: {...}. This icon does the same thing as the {...} icon.

Note: It is important to place the cursor within the element content and not to select the entire element. Selecting the entire element causes the Apply Element context to be activated instead, which allows you to replace the selected element with another element.

3.2.6 Entering attribute values

An attribute is a property of an element, and an element can have any number of attributes. Attributes have values. You may sometimes be required to enter XML data as an attribute value. In Authentic View, you enter attribute values in two ways:

- As content in the main window if the Attribute has been created to accept its value in this way
- In the Attributes Entry Helper

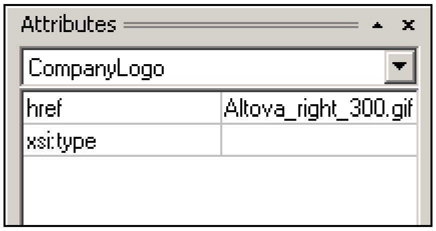
Attribute values in the main window
Attribute values can be entered as free-flowing text or as text in an input field, or as a user selection that will be mapped to an XML value. They are entered in the same way that element content is entered: see Adding document content and elements.

In such cases, the distinction between element content and attribute value is made by the StyleVision Power Stylesheet and the data is handled appropriately.

Attribute values in the Attributes Entry Helper
If you wish to enter or change an attribute value, you can also do this in the Attributes Entry Helper.

The location of the logo that is used in `NanonullOrg.xml` is stored as the value of the `href` attribute of the `CompanyLogo` element. You can change the logo to be used by doing the following:

1. Select the `CompanyLogo` element by either selecting the element or clicking the Nanonull logo. The attributes of the `CompanyLogo` element are displayed in the Attributes Entry Helper.
2. In the Attributes Entry Helper, change the value of the `href` attribute from `nanonull.gif` to `Altova_right_300.gif` (an image in the `Examples` folder).



href	Altova_right_300.gif
xsi:type	

This causes the Nanonull logo to be replaced by the Altova logo.

Note: If you are required to enter the value of an attribute, the designer of the StyleVision Power Stylesheet will, typically, include an input mechanism for this data in Authentic View.

3.2.7 Adding entities

An entity in Authentic View is typically XML data (but not necessarily), such as a single character; a text string; and even a fragment of an XML document. An entity can also be a binary file, such as an image file.

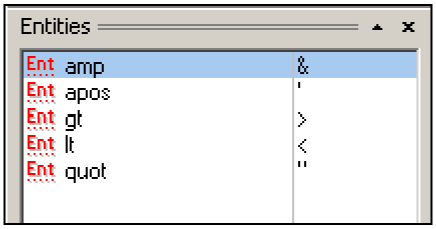
All the entities available to you are displayed in the Entities Entry Helper. To insert an entity, double-click it.

In `NanonullOrg.xml`, change the title of Joe Martin (in Marketing) to Marketing Manager Europe & Asia.

The ampersand character (&) has special significance in XML (as have the apostrophe, less than and greater than symbols, and the double quote). To insert these characters, entities are used so that they are not confused with XML-significant characters. In `NanonullOrg.xml`, entities have been declared for these characters, and are therefore displayed in the Entities Entry Helper.

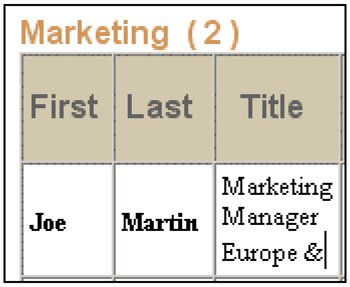
To insert the ampersand entity in the title, "Marketing Manager Europe & Asia", do the following:

1. Place the cursor where the ampersand is to be inserted.
2. Double click the entity listed as "amp".



Ent amp	&
Ent apos	'
Ent gt	>
Ent lt	<
Ent quot	"

This inserts an ampersand.


Marketing (2)

First	Last	Title
Joe	Martin	Marketing Manager Europe &

Note: The Entities Entry Helper is not context-sensitive. All available entities are displayed no matter where the cursor is positioned. This does not mean that an entity can be inserted at all locations in the document. If you are not sure, then validate the document after inserting the entity: **XML | Validate** (**F8**).

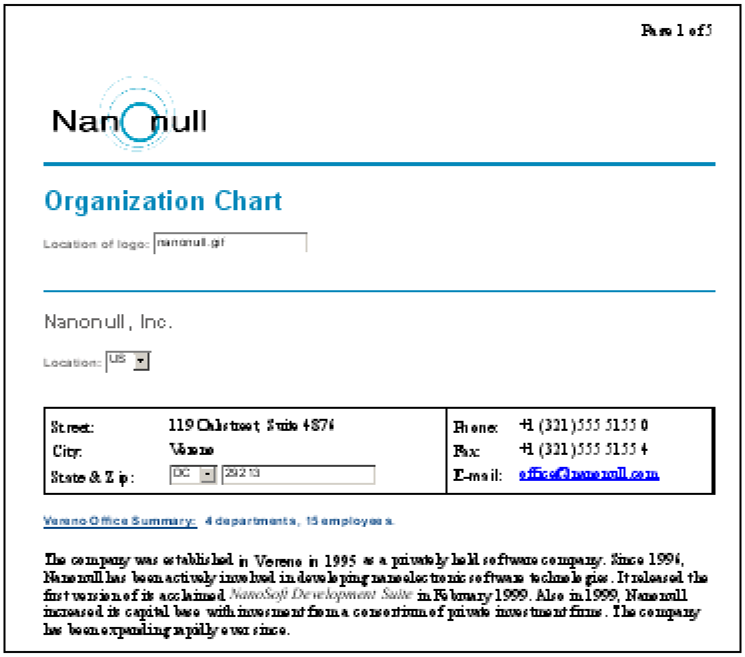
Also see: Attributes Entry Helper under Authentic View entry helpers.

3.2.8 Printing the document

A printout from Authentic View of an XML document preserves the formatting seen in Authentic View.

To print `NanonullOrg.xml`, do the following:

1. Switch to Hide Markup mode if you are not already in it. You must do this if you do not want markup to be printed.
2. Select **File | Print Preview** to see a preview of all pages. Shown below is part of a print preview page, reduced by 50%.



Page 1 of 3

Organization Chart

Location of logo: nanonull.gif

Nanonull, Inc.

Location: US

Street:	119 Oakstreet, Suite 4876	Phone:	+1 (321) 555 5155 0
City:	Vereno	Fax:	+1 (321) 555 5155 4
State & Zip:	DC 29213	E-mail:	office@nanonull.com

Vereno Office Summary: 4 departments, 15 employees.

The company was established in Vereno in 1995 as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with investment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Notice that the formatting of the page is the same as that in Authentic View.

3. To print the file, click **File | Print**.

Note that you can also print a version of the document that displays (small) markup. To do this, switch Authentic View to Show small markup mode or Show large markup mode, and then print. Both modes produce a printout that displays small markup.

XMLSpy 2005 Home Edition

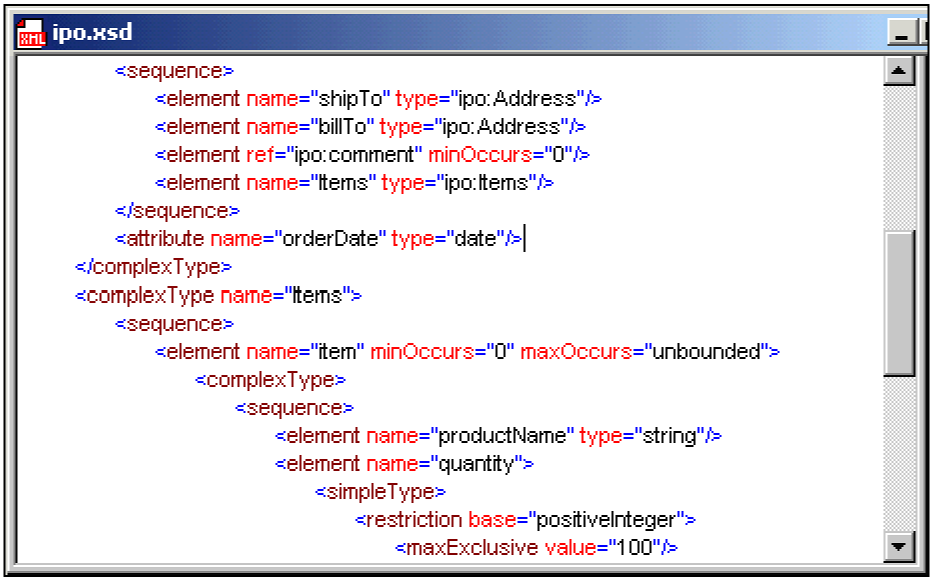
Text View

4 Text View

In Text View, you can type in your document text directly, i.e. markup and content. Text View provides a number of features to help you quickly and accurately type in your document. Among the main features are the following:

- Visual features to help you read the document more easily. These include customizable syntax-coloring (including the ability to highlight server-side VBScript or JScript code in ASP pages), indentation, customizable fonts, and text-wrapping.
- Intelligent editing features like auto-completion of tags and automatic entry of attributes and children.
- Context-sensitive entry helpers, which list allowed elements, attributes, and entities at the cursor insertion point, and allow you to insert these into the document.
- Drag-and-drop and copy-and-paste capabilities.

These features are described in more detail in the rest of this section.

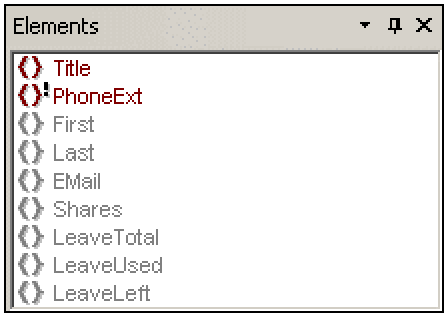


```
    <sequence>
        <element name="shipTo" type="ipo:Address"/>
        <element name="billTo" type="ipo:Address"/>
        <element ref="ipo:comment" minOccurs="0"/>
        <element name="Items" type="ipo:Items"/>
    </sequence>
    <attribute name="orderDate" type="date"/>
</complexType>
<complexType name="Items">
    <sequence>
        <element name="item" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <sequence>
                    <element name="productName" type="string"/>
                    <element name="quantity">
                        <simpleType>
                            <restriction base="positiveInteger">
                                <maxExclusive value="100"/>
```

To open the Text View of a document, click the Text button at the bottom of the Document Window or select **View | Text view**. Text view can be used to edit any text file, including non-XML documents.

4.1 Entry Helpers in Text View

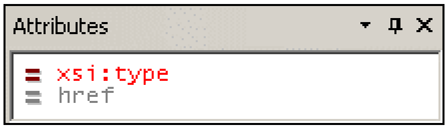
Elements Entry Helper
In Text View, elements that can be entered at the cursor point are displayed in the Elements Entry Helper in dark red. Mandatory elements are listed with an exclamation mark before the element name. Siblings of allowed elements that are themselves not allowed at the cursor point are displayed in gray. When the cursor position changes, the list in the Entry Helper changes to show only those elements that can be inserted at that point (in red) and their siblings (in gray). In the Home Edition, however, all elements are shown.



To insert an element at the cursor point, double-click the element you want to insert. The start and end tags of the element are inserted. Mandatory elements are also inserted if this option has been specified in the **Options** dialog (**Tools | Options | Editing**).

Note: In the **Options** dialog (**Tools | Options | Editing**), you can specify that mandatory child elements can be inserted when an element is inserted.

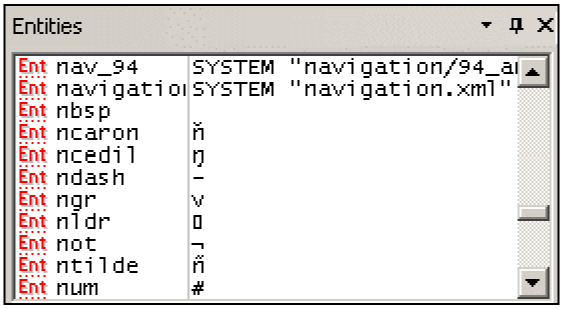
Attributes Entry Helper
In Text View, when the cursor is placed inside the start tag of an element and after a space, the attributes declared for that element become visible. Unused attributes are displayed in red, used attributes in gray. Mandatory attributes are indicated with an exclamation mark "**!**" before the name of the attribute.



To insert an attribute, double-click the required attribute. The attribute is inserted at the cursor point together with an equals-to sign and quotes to delimit the attribute value. The cursor is placed between the quotes, so you can start typing in the attribute value directly.

Note: Existing attributes, which cannot legally be added to the current element a second time, are shown in gray.

Entities Entry Helper
Any parsed or unparsed entity that is declared inline (within the XML document) or in an external DTD, is displayed in the Entities Entry Helper.



Ent	nav_94	SYSTEM "navigation/94_a
Ent	navigatio	SYSTEM "navigation.xml"
Ent	nbsp	
Ent	ncaron	ň
Ent	ncedil	ŋ
Ent	ndash	–
Ent	ngr	ν
Ent	nldr	▯
Ent	not	¬
Ent	ntilde	ñ
Ent	num	#

To insert an entity at the cursor insertion point, double-click the required entity.

Note: If you add an internal entity, you will need to save and reopen your document before the entity appears in the Entities Entry Helper.

4.2 Editing XML Documents

Syntax coloring

Syntax coloring is applied according to XML node kind, that is, depending on whether the XML node is an element, attribute, content, CDATA section, comment, or processing instruction. The text properties of these XML node kinds can be set in the Text Fonts tab of the Options dialog (**Tools | Options**).

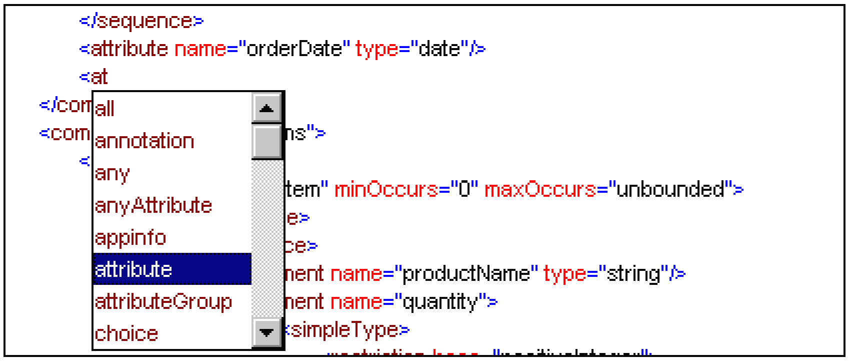
Start-tag and end-tag matching

When you place the cursor inside a start or end tag of an XML element, clicking **Ctrl+E** highlights the other member of the pair. Clicking **Ctrl+E** repeatedly enables you to switch between the start and end tags. This is another aid to locating the start and end tags of an XML element.

Intelligent Editing

If you are working with an XML document based on a DTD or XML Schema, XMLSpy 2005 provides you with various intelligent editing capabilities in Text View. These allow you to quickly insert the correct element, attribute, or attribute value according to the content model defined for the element you are currently editing. Intelligent editing typically works as follows:

1. Type **<** (the less-than character) where you want to insert an XML element. This opens a popup list containing all elements that may be legitimately inserted at that point.
2. Enter the first few characters of the element you want to insert. An element in the popup list containing those characters is highlighted.



```
        </sequence>
        <attribute name="orderDate" type="date"/>
        <at
    </com
    <com                ns">
        <
                        tem" minOccurs="0" maxOccurs="unbounded">
                        e>
                        ce>
                        nent name="productName" type="string"/>
                        nent name="quantity">
                        <simpleType>
```

3. Click on the entry with the mouse pointer or press Enter to accept the selected choice. Alternatively, use the arrow keys to highlight your selection and then click or press Enter.

The popup window also appears in the following cases:

- If you press the space bar when the cursor is between an element's tags and if an attribute is defined for that element. The popup will contain all available attributes.
- When the cursor is within the double-quotes delimiting an attribute value that has enumerated values. The popup will contain the enumerated values.
- When you type **</** (which signifies the start of a closing tag), the name of the element to be closed appears in the popup.

Auto-completion

Editing in Text View can easily result in XML documents that are not well-formed. For example, closing tags may be missing, mis-spelled, or structurally mismatched.

XMLSpy 2005 automatically completes the start and end tags of elements, as well as inserts all required attributes as soon as you finish entering the element name on your keyboard. The cursor is also automatically positioned between the start and end tags of the element, so that you can immediately continue to add child elements or contents:

```
<img src="" alt="">|</img>
```

Use the Check well-formedness command at any time to ensure that the document is well-formed. This check is also automatically performed every time you open or save a document.

Drag-and-Drop and Context Menus

You can also use drag-and-drop to move a text block to a new location, as well as right-click to directly access frequently used editing commands (such as Cut, Copy, Paste, Delete, Send by Mail, and) in a context menu.

The other commands in the context menu allow you to customize Text View.

Find and Replace

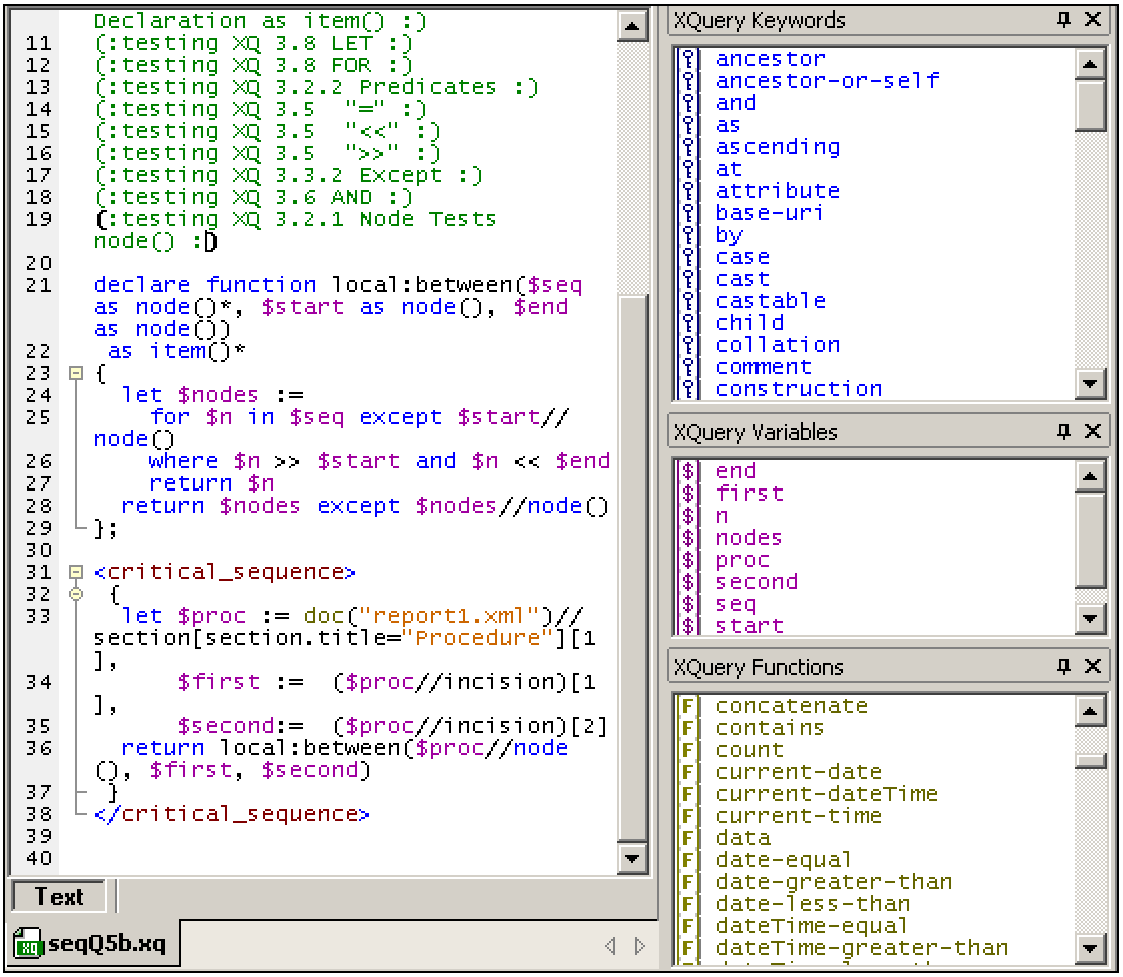
You can use the Find and Replace commands to quickly locate and change text. These commands also take regular expressions as input, thereby giving you powerful search capabilities. (See Edit **|** Find for details.)

Unlimited Undo

XMLSpy 2005 offers unlimited levels of Undo and Redo for all editing operations.

4.3 Editing XQuery Documents

In Text View, you can edit XQuery documents. The Entry Helpers, syntax coloring, and intelligent editing are different than for XML documents (*see screenshot below; line numbering and folding margins in Enterprise and Professional Editions only*). We call this mode of Text View its XQuery Mode. In addition, you can validate your XQuery document in Text View and execute the code in an XQuery document (with an optional XML file if required) using the built-in Altova XQuery Engine.



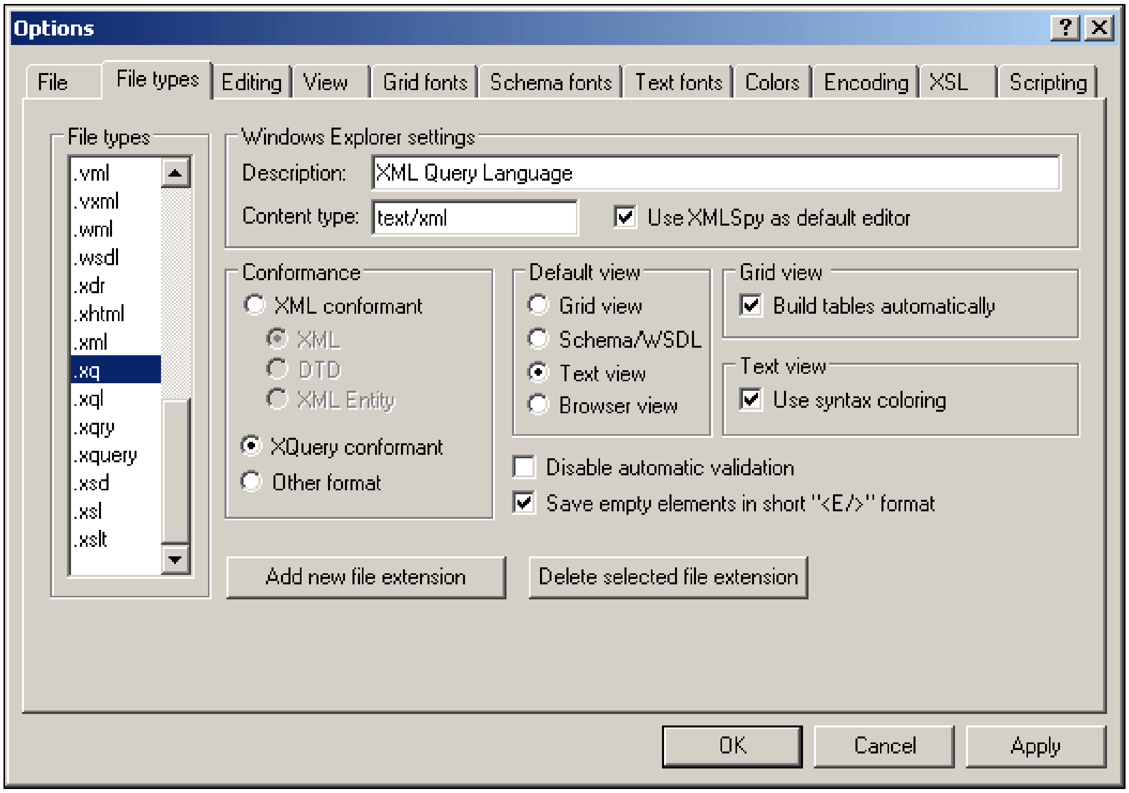
```
Declaration as item() :)
(:testing XQ 3.8 LET :)
(:testing XQ 3.8 FOR :)
(:testing XQ 3.2.2 Predicates :)
(:testing XQ 3.5  "=" :)
(:testing XQ 3.5  "<<" :)
(:testing XQ 3.5  ">>" :)
(:testing XQ 3.3.2 Except :)
(:testing XQ 3.6 AND :)
(:testing XQ 3.2.1 Node Tests
node() :)

declare function local:between($seq as node()*, $start as node(), $end as node())
 as item()*
{
  let $nodes :=
     for $n in $seq except $start//node()
     where $n >> $start and $n << $end
     return $n
  return $nodes except $nodes//node()
};

<critical_sequence>
 {
  let $proc := doc("report1.xml")//section[section.title="Procedure"][1],
        $first :=  ($proc//incision)[1],
        $second:=  ($proc//incision)[2]
  return local:between($proc//node(), $first, $second)
 }
</critical_sequence>
```

Note: XQuery files can be edited only in Text View. No other views of XQuery files are available.

For details about how the Altova XQuery Engine is implemented and will process XQuery files, see XQuery Engine Implementation.

4.3.1 Opening an XQuery Document

An XQuery document is opened automatically in XQuery Mode of Text View if it is XQuery conformant. Files that have the file extension `.xq`, `.xql`, and `.xquery` are pre-defined in XMLSpy 2005 as being XQuery conformant. You can set additional file extensions to be XQuery conformant in the File Types tab of the Options dialog (**Tools | Options**) shown below. To make a file extension XQuery conformant, add the file extension to the list of file types, and set its conformance to XQuery conformant.



File types: .vml, .vxml, .wml, .wsdl, .xdr, .xhtml, .xml, .xq (selected), .xql, .xqry, .xquery, .xsd, .xsl, .xslt

Windows Explorer settings
Description: XML Query Language
Content type: text/xml
☑ Use XMLSpy as default editor

Conformance
○ XML conformant
● XML
○ DTD
○ XML Entity
● XQuery conformant
○ Other format

Default view
○ Grid view
○ Schema/WSDL
● Text view
○ Browser view

Grid view
☑ Build tables automatically

Text view
☑ Use syntax coloring

☐ Disable automatic validation
☑ Save empty elements in short "<E/>" format

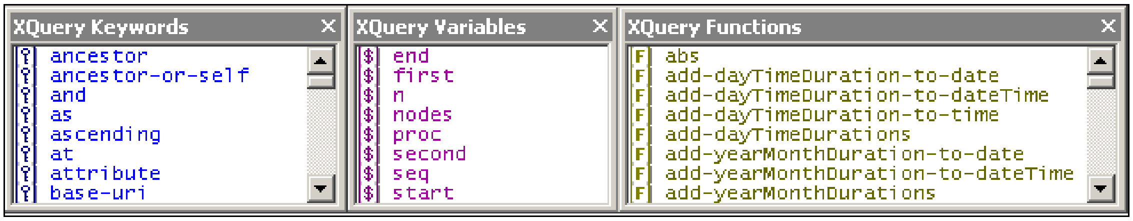
You should also make the following Windows Explorer settings:

- Description: XML Query Language
- Content type: text/xml

If you wish to use XMLSpy 2005 as the default editor for XQuery files, you should select the check box for this option.

4.3.2 XQuery Entry Helpers

There are three Entry Helpers in the XQuery Mode of Text View: XQuery Keywords (blue), XQuery Variables (purple), and XQuery Functions (olive).


XQuery Keywords: ancestor, ancestor-or-self, and, as, ascending, at, attribute, base-uri

XQuery Variables: end, first, n, nodes, proc, second, seq, start

XQuery Functions: abs, add-dayTimeDuration-to-date, add-dayTimeDuration-to-dateTime, add-dayTimeDuration-to-time, add-dayTimeDurations, add-yearMonthDuration-to-date, add-yearMonthDuration-to-dateTime, add-yearMonthDurations

The following points should be noted:

- The color of items in the three Entry Helpers are different and correspond to the syntax coloring used in the text. These colors cannot be changed.
- The listed keywords and functions are those supported by the Altova XQuery Engine.
- The variables are defined in the XQuery document itself. When a $ and a character are entered in Text View, the character is entered in the Variables Entry Helper (unless a variable consisting of exactly that character exists). As soon as a variable name that is

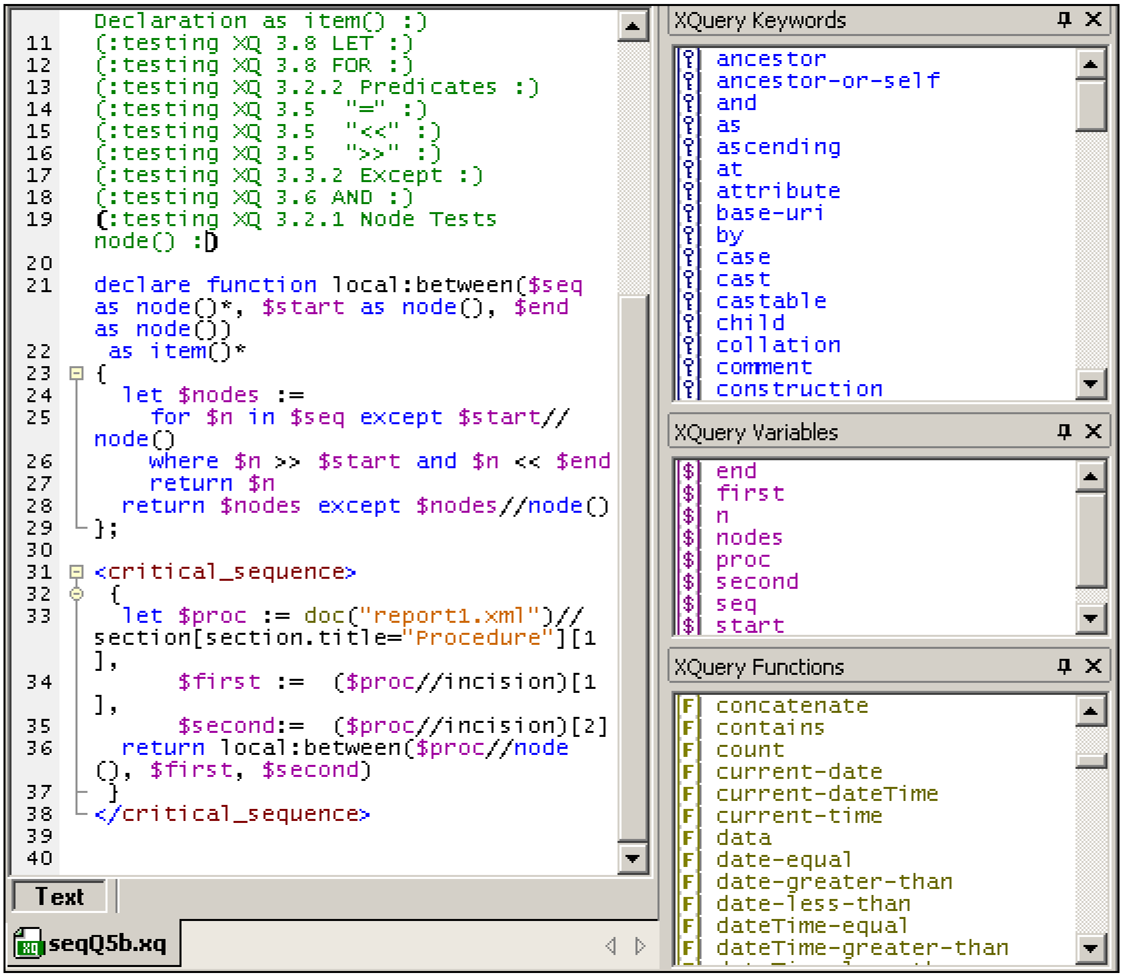
being entered matches a variable name that already exists, the newly entered variable name disappears from the Entry Helper.

- To navigate in any Entry Helper, click an item in the Entry Helper, and then use either the scrollbar, mouse wheel, or page-down and page-up to move up and down the list.

To insert any of the items listed in the Entry Helpers into the document, place the cursor at the required insertion point and double-click the item. In XQuery, some character strings represent both a keyword and a function (`empty`, `unordered`, and `except`). These strings are always entered as keywords (in blue)—even if you select the function of that name in the Functions Entry Helper. When a function appears in blue, it can be distinguished by the parentheses that follow the function name.

4.3.3 XQuery Syntax Coloring

An XQuery document can consist of XQuery code as well as XML code. The default syntax coloring for the XQuery code is described in this section. The syntax coloring for XML code in an XQuery document is the same as that used for regular XML documents. All syntax coloring (for both XQuery code and XML code) is set in the Text Fonts tab of the Options dialog (**Tools | Options**). Note that XQuery code can be contained in XML elements by enclosing the XQuery code in curly braces **{}** (*see screenshot for example; line numbering and folding margins in Enterprise and Professional Editions only*).



```
   Declaration as item() :)
(:testing XQ 3.8 LET :)
(:testing XQ 3.8 FOR :)
(:testing XQ 3.2.2 Predicates :)
(:testing XQ 3.5  "=" :)
(:testing XQ 3.5  "<<" :)
(:testing XQ 3.5  ">>" :)
(:testing XQ 3.3.2 Except :)
(:testing XQ 3.6 AND :)
(:testing XQ 3.2.1 Node Tests
node() :)

declare function local:between($seq
as node()*, $start as node(), $end
as node())
 as item()*
{
  let $nodes :=
    for $n in $seq except $start//
node()
    where $n >> $start and $n << $end
    return $n
  return $nodes except $nodes//node()
};

<critical_sequence>
 {
  let $proc := doc("report1.xml")//
section[section.title="Procedure"][1
],
      $first :=  ($proc//incision)[1
],
      $second:=  ($proc//incision)[2]
  return local:between($proc//node
(), $first, $second)
 }
</critical_sequence>
```

In XQuery code in the XQuery Mode of Text View, the following default syntax coloring is used:

- **`(: Comments, including 'smiley' delimiters, are in green :)`**
- XQuery Keywords are in blue: **`keyword`**
- XQuery Variables, including the dollar sign, are in purple: **`$start`**
- XQuery Functions, but **not** their parentheses, are in olive: **`function()`**
- Strings are in orange: **`"Procedure"`**
- All other text, such as path expressions, is black (*shown underlined below*). So:

```
for $s in doc("report1.xml")//section[section.title =
"Procedure"]
return ($s//incision)[2]/instrument
```

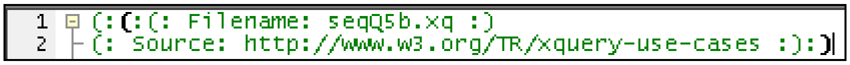
You can change these default colors and other font properties in the Text Fonts tab of the Options dialog (**Tools | Options**).

Note: In the above screenshot, one pair of colored parentheses for a comment is displayed black and bold. This is because of the bracket-matching feature (see XQuery Intelligent Editing).

4.3.4 XQuery Intelligent Editing

The XQuery Mode of Text View provides the following intelligent editing features.

Bracket-matching
The bracket-matching feature highlights the opening and closing brackets of a pair of brackets, enabling you to clearly see the contents of a pair of brackets. This is particularly useful when brackets are nested, as in XQuery comments (*see screenshot below*).



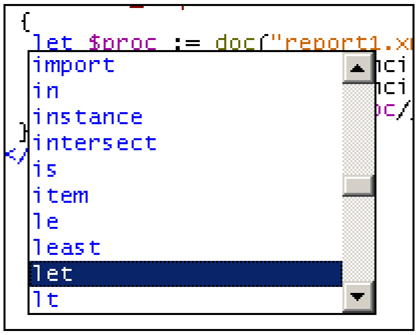
```
(:(:(: Filename: seqQ5b.xq :)
(: Source: http://www.w3.org/TR/xquery-use-cases :):)
```

Bracket-matching is activated when the cursor is placed either immediately before or immediately after a bracket (either opening or closing). That bracket is highlighted (bold black) together with its corresponding bracket. Notice the cursor position in the screenshot above.

Bracket-matching is enabled for round parentheses **`()`**, square brackets **`[]`**, and curly braces **`{}`**. The exception is angular brackets **<>**, which are used for XML tags.

Note: When you place the cursor inside a start or end tag of an XML element, clicking **Ctrl+E** highlights the other member of the pair. Clicking **Ctrl+E** repeatedly enables you to switch between the start and end tags. This is another aid to locating the start and end tags of an XML element.

Keywords
XQuery keywords are instructions used in query expressions, and they are displayed in blue. You select a keyword by placing the cursor inside a keyword, or immediately before or after it. With a keyword selected, pressing **Ctrl+Space** causes a complete list of keywords to be displayed in a pop-up menu. You can scroll through the list and double-click a keyword you wish to have replace the selected keyword.

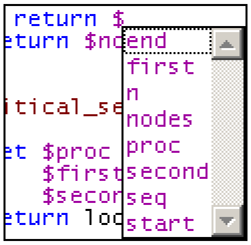


In the screenshot above, the cursor was placed in the `let` keyword. Double-clicking a keyword from the list causes it to replace the `let` keyword.

Variables

Names of variables are prefixed with the **$** sign, and they are displayed in purple. This mechanism of the intelligent editing feature is similar to that for keywords. There are two ways to access the pop-up list of all variables in a document:

- After typing a **$** character, press **Ctrl+Space**
- Select a variable and press **Ctrl+Space**. (A variable is selected when you place the cursor immediately after the **$** character, or within the name of a variable, or immediately after the name of a variable.)

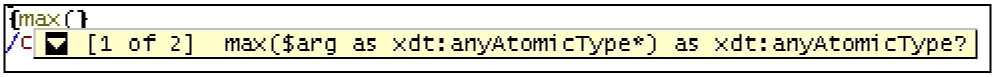


To insert a variable after the **$** character (when typing), or to replace a selected variable, double-click the variable you want in the pop-up menu.

Functions

Just as with keywords and variables, a pop-up menu of built-in functions is displayed when you select a function (displayed in olive) and press **Ctrl+Space**. (A function is selected when you place the cursor within a function name, or immediately before or after a function name. The cursor must not be placed between the parentheses that follow the function's name.) Double-clicking a function name in the pop-up menu replaces the selected function name with the function from the pop-up menu.

To display a tip containing the signature of the inserted function (*screenshot below*), place the cursor immediately after the opening parentheses and press **Ctrl+Space**.



```
{max()}
[1 of 2]  max($arg as xdt:anyAtomicType*) as xdt:anyAtomicType?
```

The downward-pointing arrowhead indicates that there is more than one function with the same name. (Such functions differ from each other in the number and type of arguments, or the return type, they take.) Clicking on the tip displays the next function, and repeatedly clicking on the tip cycles you through all the functions with that name. Alternatively, you can use the **Ctrl+Shift+Up** or **Ctrl+Shift+Down** key-combinations to move through a sequence.

4.3.5 Validation and Execution of XQuery Documents

Validating XQuery documents
To validate an XQuery document, do the following:

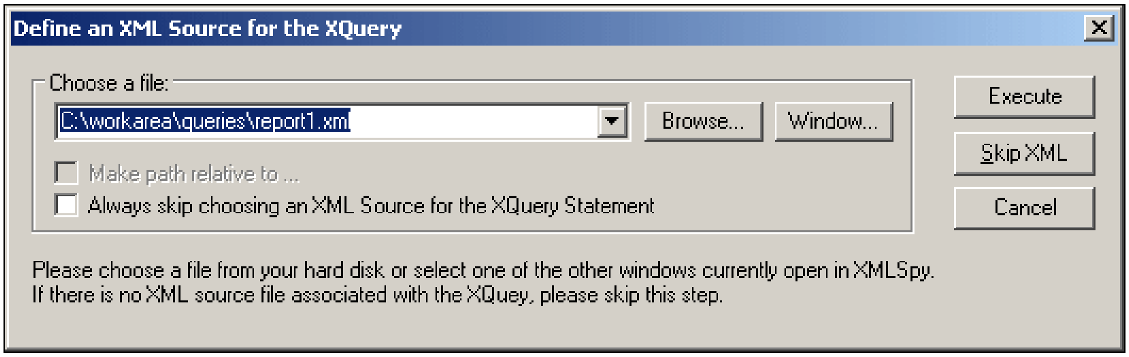
1. Make the XQuery document the active document.
2. Select **XML | Validate**, or press the **F8** key, or click the toolbar icon.

The document will be validated for correct XQuery syntax.

Executing XQuery documents
XQuery documents are executed within XMLSpy 2005 using the built-in XQuery 1.0 engine. The output is displayed in a window in XMLSpy 2005.

Typically, an XQuery document is not associated with any single XML document. This is because XQuery expressions can select any number of XML documents with the `doc()` function. In XMLSpy 2005, however, before executing individual XQuery documents you can select a source XML document for the execution. In such cases, the document node of the selected XML source is the starting context item available at the root level of the XQuery document. Paths that begin with a leading slash are resolved with this document node as its context item. To execute an XQuery document, do the following:

1. Make the XQuery document the active document.
2. Select **XSL/XQuery | XQuery Transformation** or click the toolbar icon. This pops up the Define an XML Source for the XQuery dialog.



Define an XML Source for the XQuery

Choose a file: C:\workarea\queries\report1.xml

□ Make path relative to ...

□ Always skip choosing an XML Source for the XQuery Statement

Please choose a file from your hard disk or select one of the other windows currently open in XMLSpy.
If there is no XML source file associated with the XQuey, please skip this step.

3. You can now select an XML source if you wish to assign its document node as the context item for the root level of the XQuery document. Otherwise you can skip this dialog. To select an XML file, use either the Browse button or the Window button (which lists files that are open in XMLSpy 2005 and that are in XMLSpy projects).

The result document is generated as a temporary file that can be saved to any location with the desired file format and extension.

For details about how the Altova XQuery Engine is implemented and will process XQuery files, see XQuery Engine Implementation.

XMLSpy 2005 Home Edition

Schema/WSDL Design View

5 Schema/WSDL Design View

The **Schema/WSDL Design View** is enabled for XML Schema documents. The Schema/WSDL Design View will hereafter be referred to as the Schema Design View, and it is described in detail in the following sub-section. For a description of how to create XML Schemas, see the XML Schema Tutorial.

5.1 Schema Design View

The Schema Design View itself has two types of view:

- A main **Schema Overview**, which displays all global components (global elements, complex types, etc) in a simple table.
- Views of the content models of individual global components (**Content Model View**).

Given below is a brief overview of Schema Overview and Content Model View, followed by a description of the Entry Helpers available in Schema Design View. The two sub-sections of this section contain detailed descriptions of Schema Overview and Content Model View.

Schema Overview
The Schema Overview displays a list of all the global components of the schema (global elements, complex types, etc).

You can insert, append, or delete global components, as well as modify their properties. To insert, append, or delete, use the respective buttons at the top of the Schema Overview. To modify properties, select the required component in the Schema Overview list, and edit its properties in either the entry helpers (at right of view) or the Attributespane (at bottom of view).

Note the following editing features of Schema Overview:

- You can reposition components in the Schema Overview list using drag-and-drop.
- You can navigate using the arrow keys of your keyboard.
- You can copy or move global components, attributesto a different position and from one schema to another using cut/copy-and-paste.
- Right-clicking a component opens a context menu that allows you to cut, copy, paste, delete, or edit the annotation data of that component.

Content Model View
A content model is a description of the structure and contents of an element. Global components which can have a content model (for example, elements, complex types, and model groups; but not, for example, simple types) are indicated in the Schema Overview list with a ⊞ icon to the left of the component name. Clicking on this icon opens the Content Model View for that global component. Alternatively, (i) select a component and then select the menu option **Schema design | Display Diagram,** or (ii) double-click on a component's name in the Component Navigator (which is the entry helper at top right). Note that only one content model in the schema can be open at a time. When a content model is open, you can jump to the content model of a component within the current content model by holding down **Ctrl** and double-clicking the required component.

The content model is displayed in the Content Model View as a tree (see screenshot below).).

Note the following editing features of Content Model View:

- Each level (of elements or element groups) in the tree is joined to adjacent levels with a compositor.
- Drag-and-drop functionality enables you to move tree objects (compositors, elements, element groups) around.
- You can add objects (compositors, elements, and element groups) via the context menu (right-click an object).
- You can edit the properties of an object in the Details entry helper (compositors, elements, element groups) and the Attributespane.

- The Attributes of a component are displayed in a pane at the bottom of the Main Window.

These features are explained in detail in the subsections of this section and in the tutorial.

To return to the Schema Overview, click the **Show Globals** icon or select the menu option **Schema design | Display All Globals**.

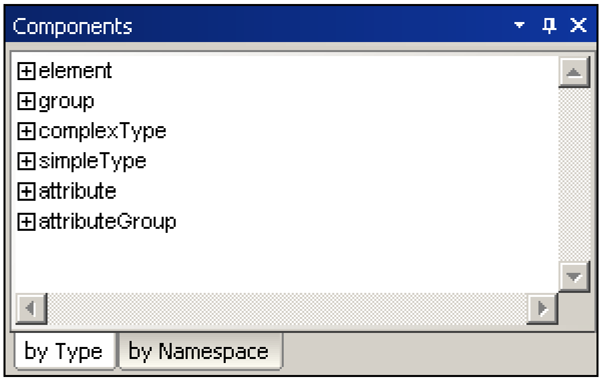
Entry Helpers in Schema View

There are three Entry Helpers in Schema/WSDL Design View: Component Navigator, Details Entry Helper, and Facets Entry Helper. These are described below.

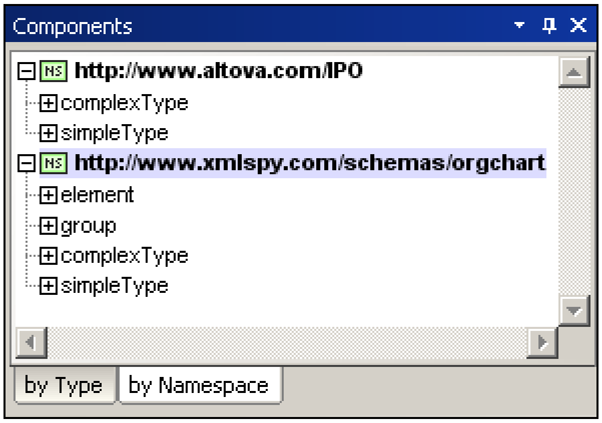
Component Navigator

The Component Navigator is an Entry Helper in **Schema/WSDL Design View.** It serves two purposes:

- To organize global components in a tree view by component type and namespace *(see screenshots below)*. This provides organized overviews of all global components.
- To enable you to navigate to and display the Content Model View of a global component—if the component has a content model. If a component does not have a content model, the component is highlighted in the Schema Overview. Global components that are included or imported from other schemas are also displayed in the Component Navigator.



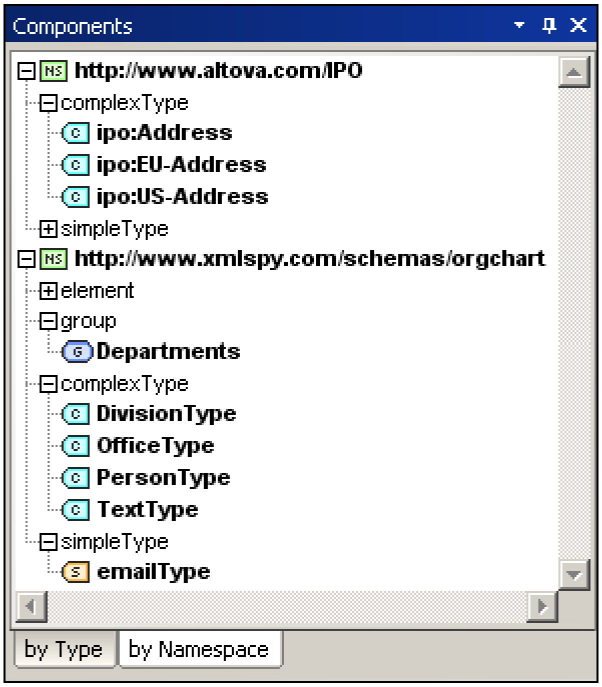
In the Type tab *(above)* global components are grouped in a tree according to their component type. In the Namespace tab *(below)*, components are organized first according to namespace and then according to component type. Note that a component type is listed in a tree only if at least one component of that type exists in the schema.



In the tree display, global components are organized into the following six groups:

- Element Declarations (Elements)
- Model Groups (Groups)
- Complex Types
- Simple Types
- Attribute Declarations (Attributes)
- Attribute Groups

Expanding a component-type group in the tree displays all the components in that group *(see screenshot below)*. This enables you to easily navigate to a required component.

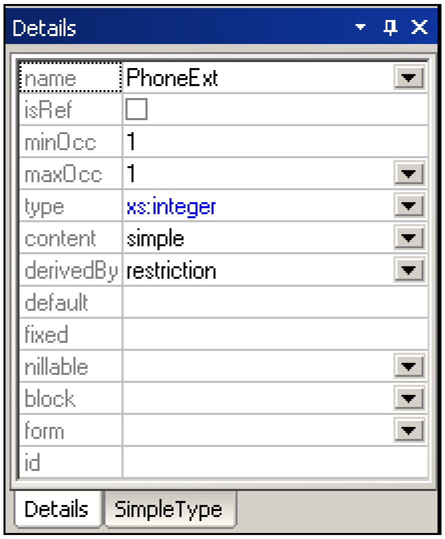


If a component has a content model (i.e. if it is an Element, Group, or Complex Type), double-clicking it will cause the content model of that component to be displayed in Content Model View (in the Main Window). If the component does not have a content model (i.e. if it is a Simple Type, Attribute, or Attribute Group), then the component is highlighted in the Schema Overview (in the Main Window).

Note: If the component is in an included or imported schema, then the included/imported schema is opened (if it is not already open), and either the component's content model is displayed in Content Model View or the component is highlighted in Schema Overview.

Details Entry Helper

The Details Entry Helper is available in **Schema/WSDL Design View**. It displays editable information about the compositor or component currently selected in the Main Window. If you are editing a schema file which contains database extensions, an additional tab with information about the DB extensions may be visible. Currently supported databases are: Oracle, SQL Server, and Tamino.



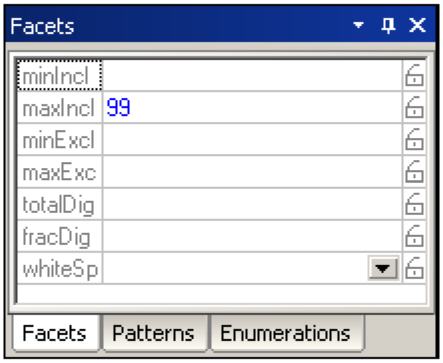
name	PhoneExt
isRef	□
minOcc	1
maxOcc	1
type	xs:integer
content	simple
derivedBy	restriction
default	
fixed	
nillable	
block	
form	
id	

To change the properties of the currently selected compositor or component, double-click the field to be edited and edit or enter text directly. If a combo box is available in the field to be edited, select the desired value; this value is entered in the field.

Changes you make via the Details Entry Helper are immediately reflected in the content model diagram.

Facets Entry Helper

The Facets Entry Helper is available in the **Schema/WSDL Design View**, and enables you to enter the values of facets, patterns, and enumerations. For examples of how to use facets and enumerations in an XML Schema, please see the relevant sections in the tutorial.



minIncl	
maxIncl	99
minExcl	
maxExc	
totalDig	
fracDig	
whiteSp	

To change facets, patterns, or enumerations in the Facets Entry Helper, do the following:

1. Select the required tab (Facets, Patterns, or Enumerations)
2. If a combo box is present, select a value from the drop-down menu. Alternatively, double-click a row, and edit or enter text directly.

Note: You can use the cut, copy and paste shortcuts (CTRL+X, CTRL+C, CTRL+V, respectively) to copy the patterns and enumerations of one component to another component. In the Facets Entry Helper, select the pattern/s or enumeration/s to copy, cut or copy the selection, then click in the Facets Entry Helper window of the target component, and paste.

5.1.1 Schema Overview

At the top level of an XML Schema document (i.e. at the level of children of the `schema` element), the following five basic components can be defined:

- Annotation
- Type definition (simple or complex)
- Declaration (element or attribute)
- Attribute group
- Model group

We call these components at the top level **global components**. The **Schema Overview** displays a list of all global components in your schema in a tabular form. Some global components (such as complex types, element declarations, and model groups) can have a content model which describes the component's structure and contents. Other global components (such as annotations, simple types, and attribute groups) do not have a content model. Those components for which content models are possible have a icon to the left of the component name. Clicking on this icon opens the Content Model View for that global component.

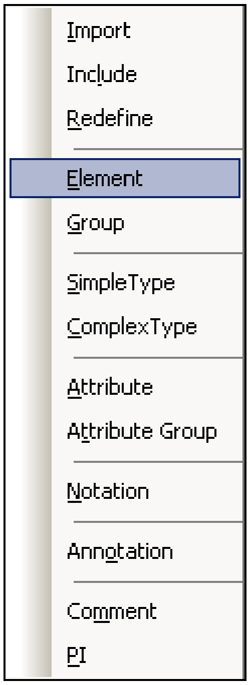
Key terms

- *Simple type and complex type.* A simple type is used to define all attributes and elements that contain only text and that have no associated attribute. A simple type, therefore, has no content model—only text (which can be restricted by the datatype). A complex type is one that has at least one child element or attribute. Declaring a child element on an element automatically assigns the element a type of complex.
- *Global and local components.* A global component can be any of the five listed above. A global component can be defined in Schema Overview, and it then immediately appears in the list of global components in Schema Overview. If the global component is a complex type, an element declaration, or a model group, you can subsequently define its content model by editing it in Content Model View. Once a global component has been defined, it can be referenced by local components. A local component is created directly within the content model of some component. Note that, in the Content Model View, a local component can be converted into a global component (via the right-click context menu).

Creating global components

To create a global component in Schema Overview, do the following:

1. Click the Insert or Append icon at the top of the Schema Overview. This pops up a menu listing the various component types (element, simple type, complex type, model group, etc).



2. Select the type of component you want. An entry of that type is created in the list of global components.
3. Enter the name of the component in the entry, and press **Enter**. The name of the new global component is added to the appropriate list/s (Elm, Grp, Com, Sim, etc) in the Component Navigator entry helper. You can edit the content model of the new global component either by double-clicking the component name in the Component Navigator or by clicking the icon to the left of the new component's name in the list of global components.

Note:

- You can also create a global component while editing in Content Model View. Right-click anywhere in the window and select **New Global | Element**.
- While editing in Content Model View, you can make a local element a global element—or even a complex type if the element has an element or attribute child. Select the local element, right-click anywhere in the window, and select **Make Global | Element** or **Make Global | Complex type**.

Deleting global components

To delete a global component, do the following:

1. Select the global component in the list of global components in the Schema Overview.
2. Press the **Delete** key, or click the **Delete** icon at the top of the Schema Overview.

Attributes of components

You can define attributes for components in either Schema Overview or Content Model View. In Schema Overview, the attributes of a component are displayed in the Attributespane at the bottom of the Schema Overview window and can be edited there.

Defining attributes for a component

To define attributes for a component, you use the Attributes pane, which is at the bottom of the Schema Overview window.

To define attributes for a global component for which attributes are allowed, do the following:

1. Select the global component in the global components list.
2. In the Attributespane, select the Attributes tab.
3. Click the Append or Insert icon at the top left of the Attribute tab.
4. From the popup that appears, select the attribute type you want to append or insert. An entry is created in the Attribute list.
5. In the newly created entry, enter the attribute's properties.

Note: You can also define attributes for global components in Content Model View: Select the global component, and then define attributes as described above.

5.1.2 Content Model View

The **Content Model View** enables you to quickly define the content model of the following three component types graphically with a few mouse clicks:

- Complex types,
- Element declarations, and
- Model groups graphically with just a few mouse clicks.

All other schema components (annotations, attribute declarations, simple types, etc) do not have a content model.

In Content Model View, the various parts of the content model are represented graphically, These parts are organized into two broad groups: **compositors** and **components**. Typically a compositor is added and then the desired child components.

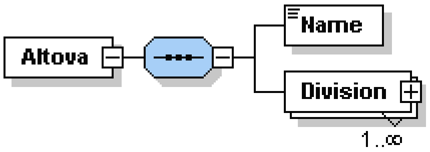
Compositors

A **compositor** defines the order in which child elements occur. There are three compositors: `sequence`, `choice`, and `all`. To insert a compositor, do the following:

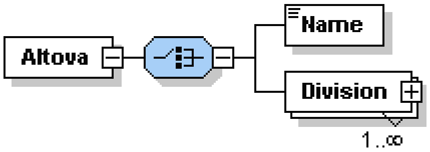
1. Right-click the element to which you wish to add child elements
2. Select **Add Child | Sequence** (or **Choice** or **All**).

The compositor is added, and will look as below:

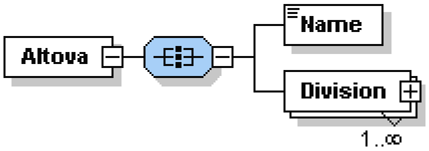
- **Sequence**



- **Choice**



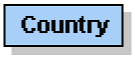
- **All**



To change the compositor, right-click the compositor and select **Change Model | Sequence** (or **Choice** or **All**). After you have added the compositor, you will need to add child element/s or a model group.

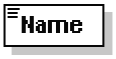
Components in the Content Model
Given below is a list of components that are used in content models. The graphical representation of each provides detailed information about the component's type and structural properties.

- Mandatory single element

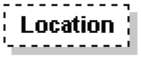
  Details: The rectangle indicates an element and the solid border indicates that the element is required. The absence of a number range indicates a single element (i.e. `minOcc=1` and `maxOcc=1`). The name of the element is `Country`. The blue color indicates that the element is currently selected; (a component is selected by clicking it). When a component is not selected, it is white.

- Mandatory single element containing parsed character data (i.e. a child text node)

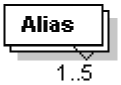
  Details: Rectangle = element; solid border = mandatory; absent number range = `minOcc=1` and `maxOcc=1`; text-line symbol at top left = a child text node (content type could be simple content (text node only) or mixed content (text and elements); in this screenshot, simple content is denoted because there is no plus sign indicating that the element can be expanded further to display a content model). Element name is `Name`.

- Single optional element

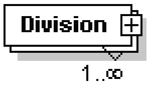
  Details: Rectangle = element; dashed border = optional; absent number range = `minOcc=0` and `maxOcc=1`. Element name is `Location`.
  Note: The context menu option **Optional** converts a mandatory element into an optional one.

- Mandatory multiple element

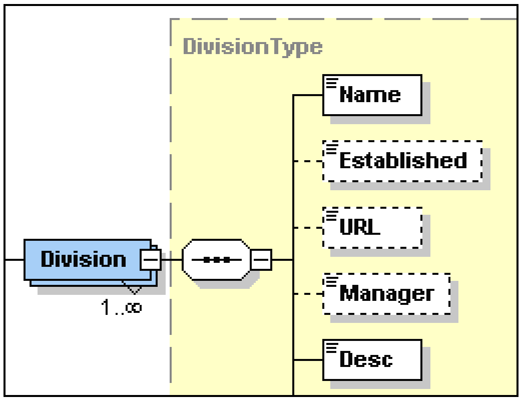
  

  Details: Rectangle = element; solid border = mandatory; number range 1..5 = `minOcc=1` and `maxOcc=5`. Element name is `Alias`.

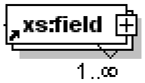
- Mandatory multiple element containing child elements



Details: Rectangle = element; solid border = mandatory; number range 1..infinity = `minOcc=1` and `maxOcc=unbounded`; plus sign = complex content (i.e. at least one element or attribute child). Element name is `Division`.
Note: The context menu option **Unbounded** changes `maxOcc` to `unbounded`. Clicking on the + sign of the element expands the tree view and shows the child elements.

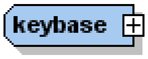


- Element referencing global element

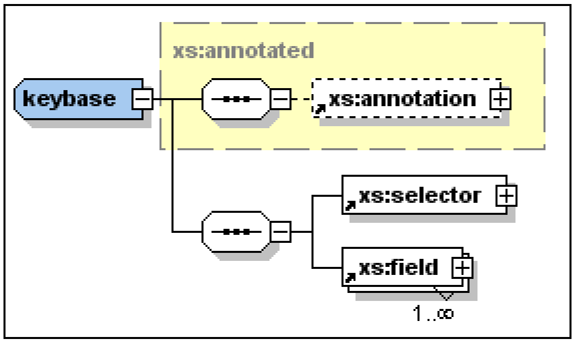


Details: Arrow in bottom-left = element referencing global element; rectangle = element; solid border = mandatory; number range 1..infinity = `minOcc=1` and `maxOcc=unbounded`; plus sign = complex content (i.e. at least one element or attribute child). Element name is `xs:field`.
Note: A global element can be referenced from within simple and complex type definitions, thus enabling you to re-use a global declaration at multiple locations in your schema. You can create a reference to a global element in two ways: (i) by entering a name for the local element that is the same as that of the global element; and (ii) by right-clicking the local element and selecting the option **Reference** from the context menu. You can view the definition of a global element by holding down **Ctrl** and double-clicking the element. Alternatively, right-click, and select **Go to Definition**. If you create a reference to an element that does not exist, the element name appears in red as a warning that there is no definition to refer to.

- Complex type



Details: The irregular hexagon with a plus sign (shown above) indicates a complex type. The complex type shown here has the name `keybase`. This symbol indicates a global complex type. A global complex type is declared in the Schema Overview, and its content model is typically defined in Content Model View. A global complex type can be used either as (i) the datatype of an element, or (ii) the base type of another complex type by assigning it to the element or complex type, respectively, in the Details entry helper (in either Content Model View or in Schema Overview).

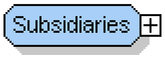


The `keybase` complex type shown above was declared in Schema Overview with a base type of `xs:annotated`. The base type is displayed as a rectangle with a dashed gray border and a yellow background color. Then, in Content Model View, the child elements `xs:selector` and `xs:field` were created. (Note the tiny arrows in the bottom left corner of the `xs:selector` and `xs:field` rectangles. These indicate that both element reference global elements of those names.)

A local complex type is defined directly in Content Model View by creating a child element or attribute for an element. There is no separate symbol for local complex types.

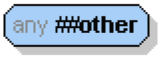
Note: The base type of a content model is displayed as a rectangle with a dashed gray border and a yellow background color. You can go to the content model of the base type by double-clicking its name.

- Model group



Details: The irregular octagon with a plus sign (shown above) indicates a model group. A model group allows you to define and reuse element declarations.
Note: When the model group is declared (in Schema Overview) it is given a name. You subsequently define its content mode (in Content Model View) by assigning it a child compositor that contains the element declarations. When the model group is used, it is inserted as a child, or inserted or appended within the content model of some other component (in Content Model View).

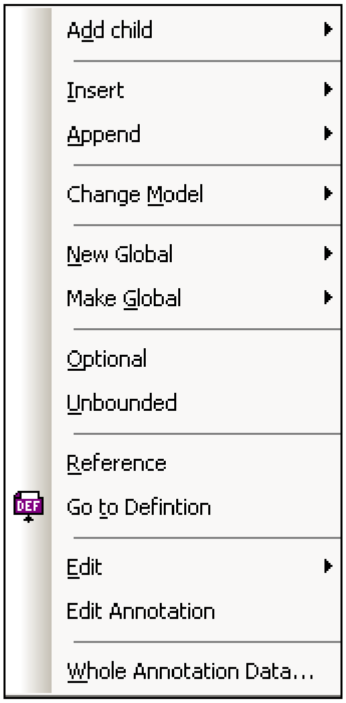
- Wildcards



Details: The irregular octagon with `any` at left indicates a wildcard.

- *Note:* Wildcards are used as placeholders to allow elements not specified in the schema or from other namespaces. `##other` = elements can belong to any namespace other than the target namespace defined in the schema; `##any` = elements can belong to any namespace; `##targetNamespace` = elements must belong to the target namespace defined in the schema; `##local` = elements cannot belong to any namespace; *`anyURI`* = elements belong to the namespace you specify.

Other editing operations in Content Model View
Editing operations in Content Model View are carried out via the context menu (*screenshot below*) that appears when you right-click within Context Model View. A description of the operations are given below.



Adding child compositors/components and inserting/appending compositors/components

1. Right-click the compositor or component. This opens the context menu (with only the allowed operations enabled).
2. Select the required operation from the context menu.

Changing a compositor

1. Right-click the compositor you want to change.
2. Select the context menu option **Change Model** and, from the sub-menu, select the compositor to which you want to change. (The currently selected compositor is checked.) If a compositor is not allowed at that point, it is grayed out.

Creating global components

- To create a new global component, right-click anywhere in Content Model View, select **New Global**, and, from the sub-menu, the required component.
- To make a local element a global element or global complex type, right-click the local element, select **Make Global**, and, from the sub-menu, select either **Element** or **Complex type**. If any of these components cannot legally be created, then it is grayed out.

Changing the occurrence definition

You can toggle the minimum and maximum occurrences values of a compositor between `0` and `1` (for `minOccurs`) and `1` and `unbounded` (for `maxOccurs`), respectively. Do this as follows:

1. Right-click the compositor or component for which the occurrence value has to be changed.

2. Select the context menu option **Optional** to toggle the `minOccurs` value between `0` and `1`, and/or select **Unbounded** to toggle the `maxOccurs` value between `1` and `unbounded`. If the options are selected (values are `0` and `unbounded`, respectively), then a check mark appears to the left of the respective menu item.

Toggling between local definition and global definition

If a global element exists that has the same name as a local element, then you can toggle between referencing the global definition and using the local definition. Do this as follows:

1. Right-click the element.
2. Select the context menu option **Reference**. If the global element is referenced, then the menu item is checked. If the local definition is used, the **Reference** item in the menu is not checked.

Jumping to another definition

When you are within a content model, you can jump to the definition of any global component that is contained in that content model. Do this as follows:

1. Right-click the global component. The global component could be the yellow rectangle of a base type; an element that references a global element; or a model group.
2. Select the context menu option **Go to Definition**. This opens the Content Model View of that global component.

Alternatively, double-click the name of the base type, or press Ctrl and double-click the referencing element or the model group.

Editing element names

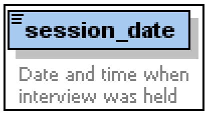
1. Right-click the element.
2. Select the context menu option **Edit | Name** and edit the name.

Alternatively, double-click the element name, and type in the change.

Creating and editing documentation for a compositor or component

You can add documentation to individual compositors and components as a guide for schema editors. Do this as follows:

1. Right-click the compositor or component.



2. Select the context menu option **Edit Annotation**. This highlights the documentation space below the compositor/component, in which you can enter descriptive text about the compositor or component. In Text View, the `annotation` and `annotation/documentation` elements will have been created and the `documentation` element will contain the descriptive text you enter.

Alternatively, you can right-click the compositor or component and select Whole Annotation Data. In the Annotation dialog thta pops up, you can append or insert a documentation item and enter content for it.

In order to edit pre-existing documentation text, you can use any of the two methods described above, but a quicker method is to double-click the annotation in the diagram and edit directly.

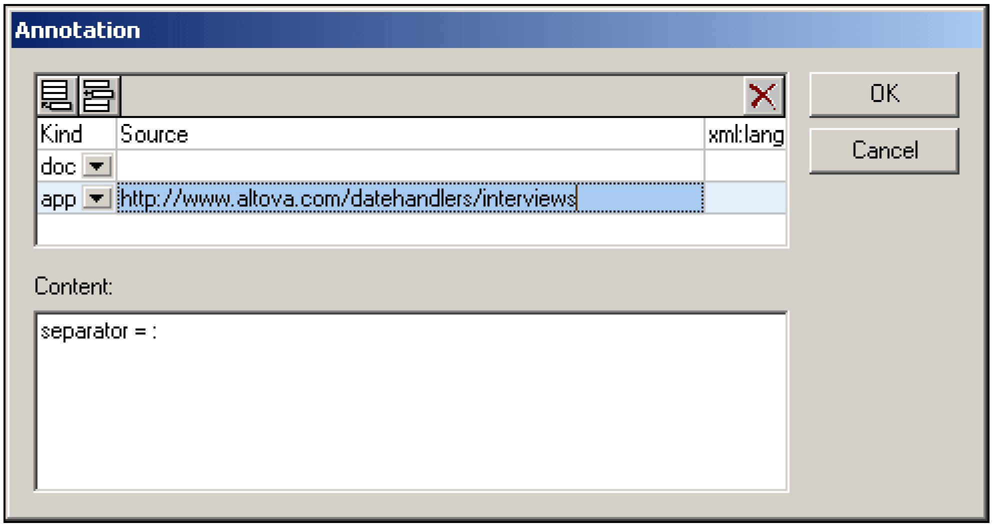
Creating and editing application info for a compositor or component

1. Right-click the compositor or component.

session_date

Date and time when interview was held

2. Select the context menu option **Whole Annotation Data**. This pops up the Annotation dialog box (*see screenshot below*). If annotation (either documentation or appinfo) exists for that element, then this is indicated by a corresponding row in the dialog.



Kind	Source	xml:lang
doc		
app	http://www.altova.com/datehandlers/interviews	

separator = :

3. To create an `appinfo` element, click the Append or Insert icon at top left to append or insert a new row, respectively.
4. In the Kind field of the new row, select the app option from the dropdown menu.
5. In the Content pane of the dialog, enter the script or info that you want to have processed by a processing application.
6. Optionally, in the Source field, you can enter a source URI where further information can be made available to the processing application.

About XML Schema annotations

XML Schema annotations are held in the `annotation` element. There are two types of annotation:, both of which are elements of the annotation element:

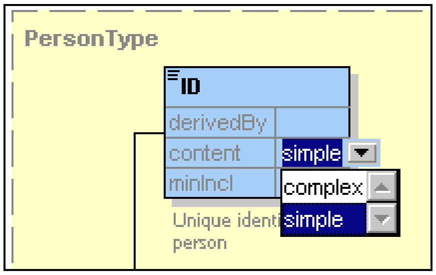
- compositor or component documentation, which contains information that could be useful for editors of the schema and is contained in the `documentation` child element of `annotation`.
- application information, which allows you to insert script or information that a processing application may use; this information is contained in the `appinfo` child element of `annotation`..

Given below is the text of an annotation element. It is based on the example in the description of creating documentation and application information given above.

```
<xs:element name="session_date" type="xs:dateTime" nillable="true">
   <xs:annotation>
      <xs:documentation>Date and time when interview was
held</xs:documentation>
      <xs:appinfo
source="http://www.altova.com/datehandlers/interviews">separator =
:</xs:appinfo>
   </xs:annotation>
</xs:element>
```

Changing component properties directly in the content model
If the Content Model View is configured so that components are displayed with property descriptor lines (additional information about components) in the component box, then you can edit this information and so change the properties of components. The property descriptor lines you have defined can be turned on and off by clicking the Add Predefined Details ▦ toolbar icon. You can toggle between a view containing the defined properties and a view not containing them. To edit component properties, do the following:

- Double-click the (component's) information field that you want to edit, and start entering or editing data. If a predefined option is available, then a drop-down list can be opened and the appropriate entry selected. Otherwise simply enter the required value.



- Confirm your entry by pressing **Return**. The Details entry helpers will be updated to reflect your changes.

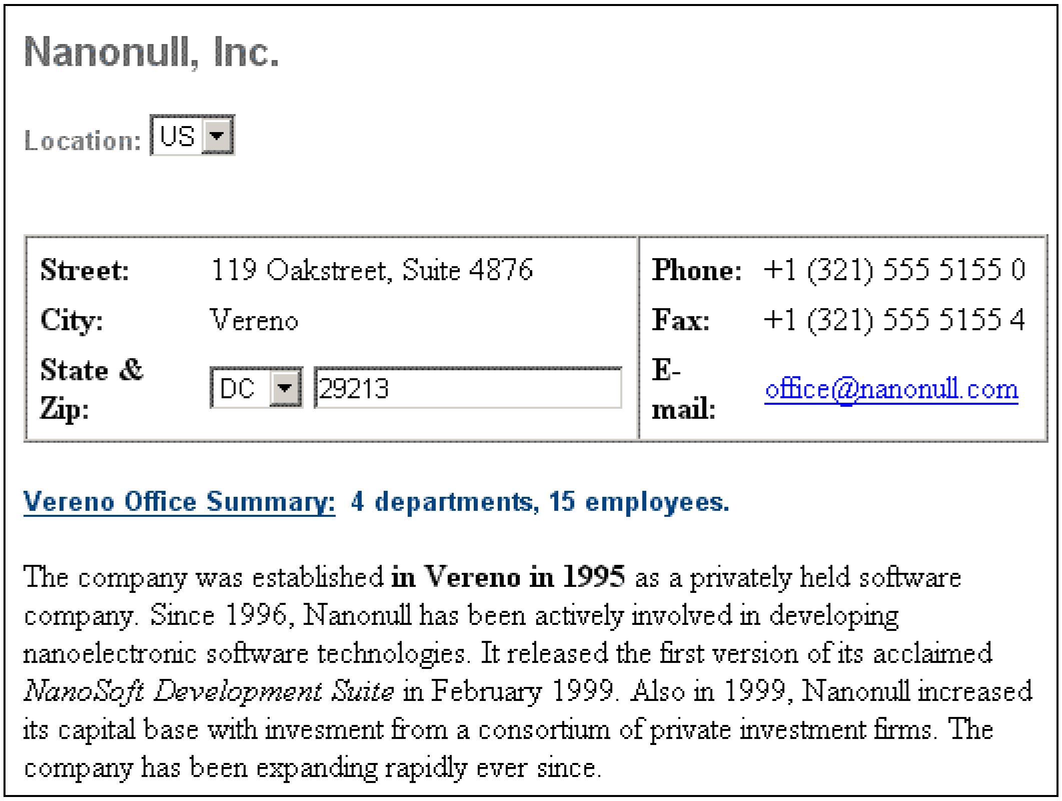
Alternatively, you can edit a component's properties in the Details entry helper, and changes will be reflected in the placeholder fields—if these are configured to be displayed.

XMLSpy 2005 Home Edition

Authentic View

6 Authentic View

Authentic View (see screenshot below) is a graphical representation of your XML document. It enables XML documents to be displayed **without** markup and **with** appropriate formatting and data-entry features such as input fields, combo boxes, and radio buttons. Data that the user enters in Authentic View is entered into the XML file.



Nanonull, Inc.

Location: US ▾

Street:	119 Oakstreet, Suite 4876	**Phone:**	+1 (321) 555 5155 0
City:	Vereno	**Fax:**	+1 (321) 555 5155 4
State & Zip:	DC ▾ 29213	**E-mail:**	office@nanonull.com

Vereno Office Summary: 4 departments, 15 employees.

The company was established **in Vereno in 1995** as a privately held software company. Since 1996, Nanonull has been actively involved in developing nanoelectronic software technologies. It released the first version of its acclaimed *NanoSoft Development Suite* in February 1999. Also in 1999, Nanonull increased its capital base with invesment from a consortium of private investment firms. The company has been expanding rapidly ever since.

Authentic View enables you to edit XML documents and databases (DBs) in an easy-to-use graphical user interface (GUI).

To be able to view and edit an XML document in Authentic View, the XML document must be associated with a **StyleVision Power Stylesheet,** which is created in Altova's StyleVision 2005 product. A StyleVision Power Stylesheet (`.sps` file) is, in essence, an XSLT stylesheet. It specifies an output presentation for an XML file that also includes data-entry mechanisms. Authentic View users can, therefore, write data back to the XML file or DB. A StyleVision Power Stylesheet is based on a schema and is specific to it. If you wish to use a StyleVision Power Stylesheet to edit an XML file in Authentic View, you must use a StyleVision Power Stylesheet that is based on the same schema as that on which the XML file is based.

Using Authentic View

- If an XML file is open, you can switch to Authentic View by clicking the **Authentic** button at the bottom of the Document Window. If a StyleVision Power Stylesheet is not already assigned to the XML file, you will be prompted to assign one to it. You must use a StyleVision Power Stylesheet that is based on the same schema as the XML file.
- A new XML file is created and displayed in Authentic View by selecting the **File | New** command and then clicking the "Select a StyleVision Stylesheet ..." button. This new file

is a template file associated with the StyleVision Power Stylesheet you open. It can have a variable amount of starting data already present in it. This starting data is contained in an XML file (a Template XML File) that may optionally be associated with the StyleVision Power Stylesheet. After the Authentic View of an XML file is displayed, you can enter data in it and save the file.

This section provides:

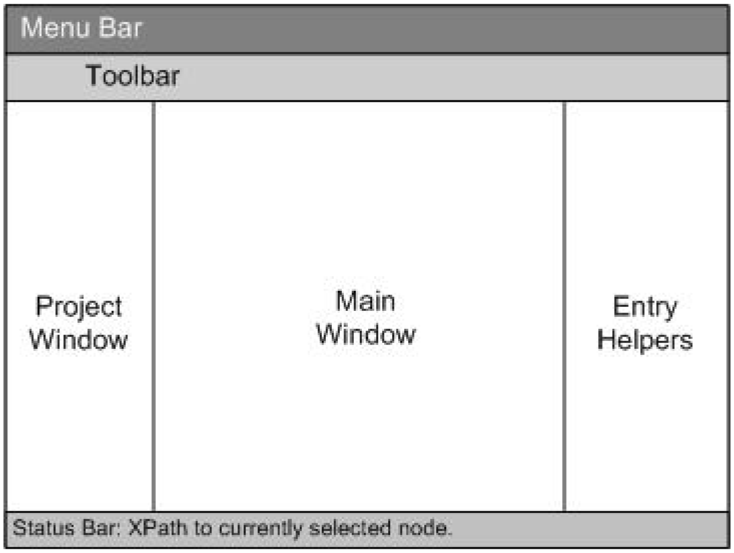
- An overview of the interface
- A description of the toolbar icons specific to Authentic View
- A description of viewing modes available in the main Authentic View window
- A description of the Entry Helpers and how they are to be used
- A description of the context menus available at various points in the Authentic View of the XML document
- A detailed description of how to use various Authentic View features

Additional sources of Authentic View information are:

- An Authentic View Tutorial, which shows you how to use the Authentic View interface. This tutorial is available in the documentation of the Altova XMLSpy 2005 and Altova Authentic 2005 Desktop Edition products (see the Tutorials section), as well as online.
- For a detailed description of Authentic View menu commands, see the User Reference section of your product documentation.

6.1 Overview of the GUI

Authentic View has a menu bar and toolbar running across the top of the window, and three areas that cover the rest of the interface: the Project Window, Main Window, and Entry Helpers Window. These areas are shown below.



Menu bar
The menus available in the menu bar are described in detail in the User Reference section of your product documentation.

Toolbar
The symbols and icons displayed in the toolbar are described in the section, Authentic View toolbar icons.

Project window
You can group XML, XSL, HTML schema, and Entity files together in a project. To create and modify the list of project files, use the commands in the Project menu (described in the User Reference section of your product documentation). The list of project files is displayed in the Project window. A file in the Project window can be accessed by double-clicking it.

Main window
This is the window in which the XML document is displayed and edited. It is described in the section, Authentic View main window.

Entry helpers
There are three entry helper windows in this area: Elements, Attributes, and Entities. What entries appear in these windows (Elements and Attributes Entry Helpers) are context-sensitive, i.e. it depends on where in the document the cursor is. You can enter an element or entity into the document by double-clicking its entry helper. The value of an attribute is entered into the value field of that attribute in the Attributes Entry Helper. See the section Authentic View Entry Helpers for details.

Status Bar
The Status Bar displays the XPath to the currently selected node.

Context menus

These are the menus that appear when you right-click in the Main Window. The available commands are context-sensitive editing commands, i.e. they allow you to manipulate structure and content relevant to the selected node. Such manipulations include inserting, appending, or deleting a node, adding entities, or cutting and pasting content.

6.2 Authentic View toolbar icons

Icons in the Authentic View toolbar are command shortcuts. Some icons will be already familiar to you from other Windows applications or your Altova product, others might be new to you. This section describes icons unique to Authentic View.

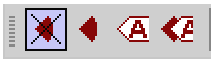
In the description below, related icons are grouped together.

Switching to Authentic View

If the XML document **is linked** to a StyleVision Power Stylesheet, **View | Authentic view** switches to Authentic View from another view.
If the document **is not linked** to a StyleVision Power Stylesheet, a dialog is displayed that asks you to link the document to a StyleVision Power Stylesheet. If, when you try to switch to Authentic View, you receive a message saying that a temporary (temp) file could not be created, contact your system administrator. The system administrator must change the default Security ID for "non-power users" to allow them to create folders and files.

Show/hide XML markup

In Authentic View, the tags for all, some, or none of the XML elements or attributes can be displayed, either with their names (large markup) or without names(small markup). The four markup icons appear in the toolbar, and the corresponding commands are available in the **Authentic** menu.



Hide markup. All XML tags are hidden except those which have been collapsed. Double-clicking on a collapsed tag (which is the usual way to expand it) in Hide markup mode will cause the node's content to be displayed and the tags to be hidden.

Show small markup. XML element/attribute tags are shown without names.

Show large markup. XML element/attribute tags are shown with names.

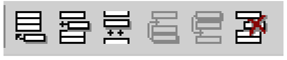
Show mixed markup. In the StyleVision Power Stylesheet, each XML element or attribute can be specified to display (as either large or small markup), or not display at all, in mixed markup mode. In mixed markup mode, therefore, the Authentic View user sees a customized markup. Note, however, that this customization is created by the person who has designed the StyleVision Power Stylesheet.

Editing dynamic table structures

Rows in a **dynamic SPS table** are repetitions of a data structure. Each row represents an occurrence of a single element. Each row, therefore, has the same XML substructure as the next.

The dynamic table editing commands manipulate the rows of a dynamic SPS table. That is, you can modify the number and order of the element occurrences. You cannot, however, edit the columns of a dynamic SPS table, since this would entail changing the substructure of individual element occurrences.

The icons for dynamic table editing commands appear in the toolbar, and are also available in the **Authentic** menu.



Append row to table

Insert row in table

Duplicate current table row (i.e. cell contents are duplicated)

Move current row up by one row

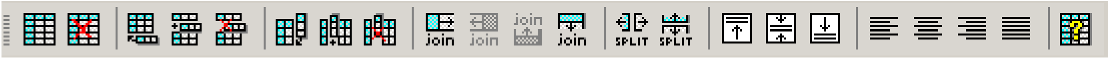
Move current row down by one row

Deletes the current row

Note: These commands apply only to **dynamic SPS tables**. They should not be used inside static SPS tables. The various types of tables used in Authentic View are described in the Using tables in Authentic View section of this documentation.

Creating and editing XML tables

You can insert your own tables should you want to present your data as a table. Such tables are inserted as XML tables. You can modify the structure of an XML table, and format the table. The icons for creating and editing XML tables are available in the toolbar, and are shown below. They are described in the section XML table editing icons.



The commands corresponding to these icons are **not available as menu items**. Note also that for you to be able to use XML tables, this function must be enabled and suitably configured in the StyleVision Power Stylesheet.

A detailed description of the types of tables used in Authentic View and of how XML tables are to be created and edited is given in the Using tables in Authentic View section of the documentation.

Text formatting icons

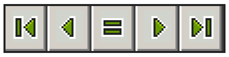
Text in Authentic View is formatted by applying to it an XML element or attribute that has the required formatting. If such formatting has been defined, the designer of the StyleVision Power Stylesheet can provide icons in the Authentic View toolbar to apply the formatting.

To apply text formatting using a text formatting icon, highlight the text you want to format, and click the appropriate icon.

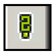
Define Entities icon

This icon opens the Define Entities dialog, which allows the Authentic View user to define entities that can then be used in the XML document.

DB Row Navigation icons

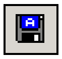


The arrow icons are, from left to right, Go to First Record in the DB; Go to Previous Record; Open Go to Record # dialog; Go to Next Record; and Go to Last Record..



This icon opens the Edit Database Query dialog in which you can enter a query. Authentic View displays the queried record/s.

Save XML icon



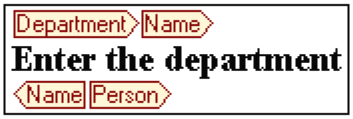
Saves changes made in Authentic Preview to the Working XML File.

6.3 Authentic View main window

There are four viewing modes in Authentic View: Large Markup; Small Markup; Mixed Markup; and Hide All Markup. These modes enable you to view the document with varying levels of markup information.

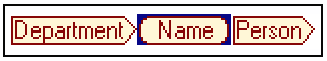
To switch between modes, use the commands in the Authentic menu or the icons in the toolbar (see the previous section, Authentic View toolbar icons).

Large markup
This shows the start and end tags of elements and attributes with the element/attribute names in the tags:



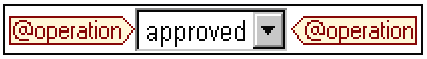
Enter the department

The element `Name` in the figure above is **expanded**, i.e. the start and end tags, as well as the content of the element, are shown. An element/attribute can be **contracted** by double-clicking either its start or end tag:



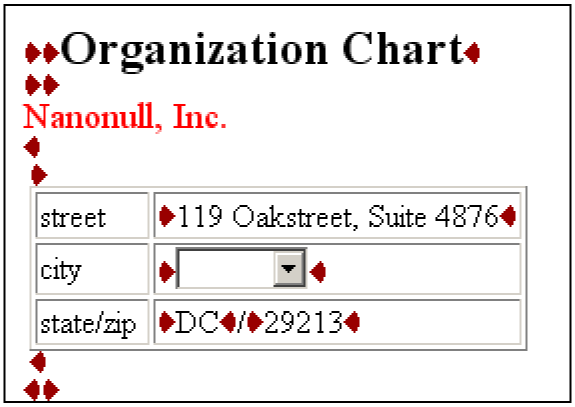
To expand the contracted element/attribute, double-click the contracted tag.

In large markup, attributes are recognized by the symbol `@` in the start and end tags of the attribute:



approved

Small markup
This shows the start and end tags of elements/attributes without names:

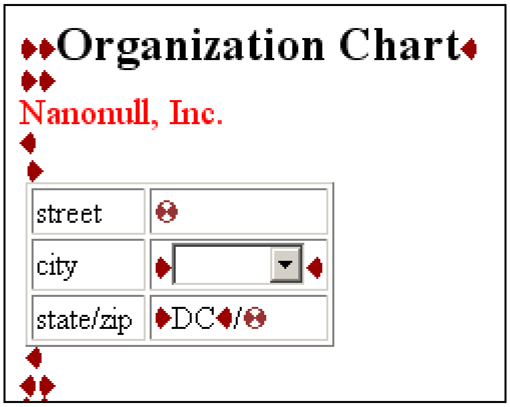


Organization Chart

Nanonull, Inc.

street	119 Oakstreet, Suite 4876
city	
state/zip	DC/29213

To contract and expand an element/attribute, double-click the appropriate tag. The example below shows two contracted elements in the table:

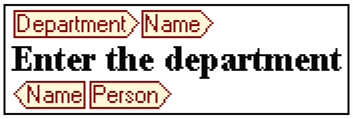


Mixed markup
Mixed markup shows a customized level of markup. The person who has designed the StyleVision Power Stylesheet can specify either large markup, small markup, or no markup for individual elements/attributes in the document. The Authentic View user see this customized markup in mixed markup viewing mode.

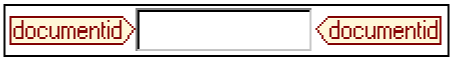
Hide all markup
All XML markup is hidden. Since the formatting seen in Authentic View is the formatting of the printed document, this viewing mode is a WYSIWYG view of the document.

Content display
In Authentic View, content is displayed in two ways:

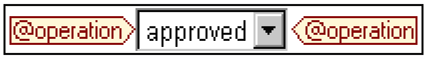
- Plain text. You type in the text, and this text becomes the content of the element or the value of the attribute.



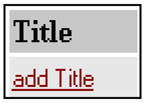
- Data-entry devices. The display contains either an input field (text box), a multiline input field, combo box, check box, or radio button. In the case of input fields and multiline input fields, the text you enter in the field becomes the XML content of the element or the value of the attribute.



In the case of the other data-entry devices, your selection produces a corresponding XML value, which is specified in the StyleVision Power Stylesheet. Thus the selection "approved" in the display example below could map to an XML value of "1", or to "approved", or anything else; while "not approved" in the display could map to "0", or "not approved", or anything else.



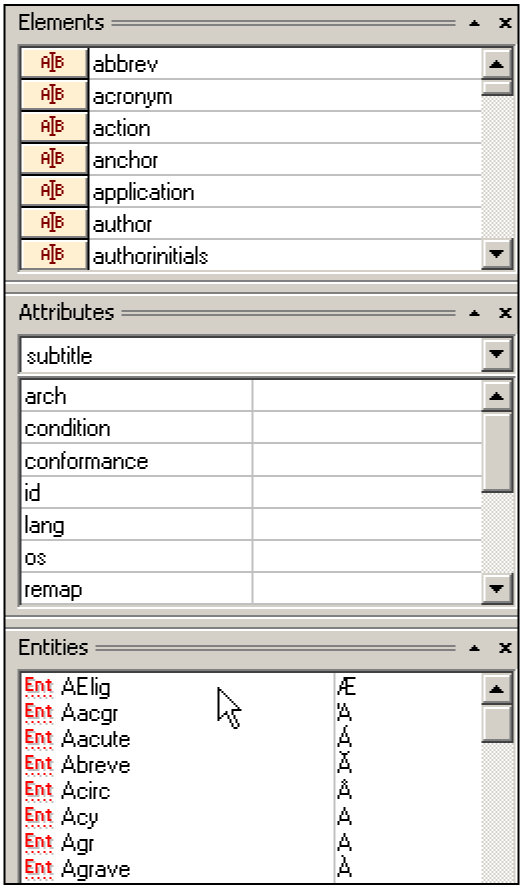
Optional nodes
When an element or attribute is **optional** (according to the referenced schema), a prompt of type "add [*element/attribute*]" is displayed:



Clicking the prompt, adds the element, and places the cursor for data entry. If there are multiple optional nodes, the prompt "add..." is displayed. Clicking the prompt, displays a menu of the optional nodes.

6.4 Authentic View entry helpers

There are three entry helpers in Authentic View: for Elements, Attributes, and Entities. They are displayed as windows down the right side of the Authentic View interface.



The Elements and Attributes Entry Helpers are context-sensitive, i.e. what appears in the entry helper depends on where the cursor is in the document. The entities displayed in the Entities Entry Helper are not context-sensitive; all entities allowed for the document are displayed no matter where the cursor is.

Each of the entry helpers is described separately below.

Elements Entry Helper
The Elements Entry Helper lists the elements that can be appended after, inserted before, inserted within, applied to (i.e. replace), the selected element or text range in Authentic View. What you can do with an element listed in the Entry Helper is indicated by the icon to the left of the element name in the Entry Helper. The icons that occur in the Elements Entry Helper are listed below, together with an explanation of what they mean.

To use an element from the Entry Helper, click its icon.

Append After Element
The element in the Entry Helper is appended after the selected element. Note that it is appended at the correct hierarchic level. For example, if your cursor is inside a `//sect1/para` element, and you append a `sect1` element, then the new `sect1` element will be appended not as a following sibling of

`//sect1/para` but as a following sibling of the `sect1` element that is the parent of that `para` element.

Insert Before Element
The element in the Entry Helper is inserted before the selected element. Note that, just as with the Append After Element command, the element is inserted at the correct hierarchic level.

Insert Element
An element from the Entry Helper can also be inserted **within an element**. When the cursor is placed within an element, then the allowed child elements of that element can be inserted. Note that allowed child elements can be part of an elements-only content model as well as a mixed content model (text plus child elements).

An allowed child element can be inserted either when a text range is selected or when the cursor is placed as an insertion point within the text.

- When a text range is selected and an element inserted, the text range becomes the content of the inserted element.
- When an element is inserted at an insertion point, the element is inserted at that point.

After an element has been inserted, it can be cleared by clicking either of the two Clear Element icons that appear (in the Elements Entry Helper) for these inline elements. Which of the two icons appears depends on whether you select a text range or place the cursor in the text as an insertion point (see below).

Apply Element
If you select an element in your document (by clicking either its start or end tag in the Show large markup view), this icon indicates that the element in the Entry Helper can be applied to the selected (original) element. The applied element *replaces* the original element.

- If the applied element has a **child element with the same name** as a child of the original element and an instance of this child element exists in the original element, then the child element of the original is retained in the new element's content.
- If the applied element has **no child element with the same name** as that of an instantiated child of the original element, then the instantiated child of the original element is appended as a sibling of any child element or elements that the new element may have.
- If the applied element has **a child element for which no equivalent exists** in the original element's content model, then this child element is not created directly but Authentic View offers you the option of inserting it.

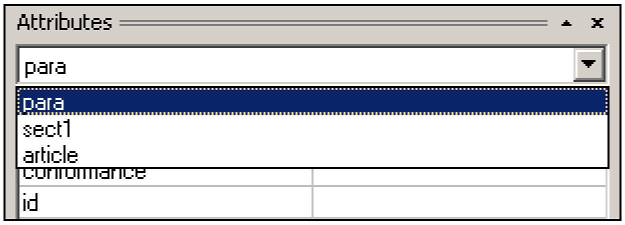
If a text range is selected rather than an element, applying an element to the selection will create the applied element at that location with the selected text range as its content. Applying an element when the cursor is an insertion point is not allowed.

Clear Element (when range selected)
This icon appears to the left of the inline element within which the selected text range is. Clicking the icon clears that inline element but not the selected text range.

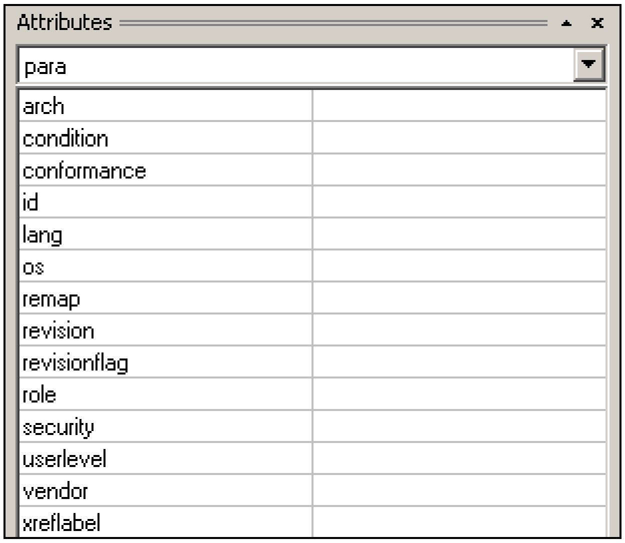
{..} **Clear Element (when insertion point selected)**
This icon appears to the left of the inline element within which the cursor is. Clicking the icon clears that inline element but not its contents.

Attributes Entry Helper
The Attributes Entry Helper consists of a drop-down combo box and a list of attributes. The element that you have selected (you can click the start or end tag, or place the cursor anywhere in the element content to select it) appears in the combo box.

The Attributes Entry Helper shown in the figures below has a `para` element in the combo box. Clicking the arrow in the combo box drops down a list of all the `para` element's **ancestors up to the document's root element**, which in this case is `article`.



Below the combo box, a list of valid attributes for that element is displayed, in this case for `para`. If an attribute is mandatory on a given element, then it appears in bold. (In the example below, there are no mandatory attributes; they are all optional.)



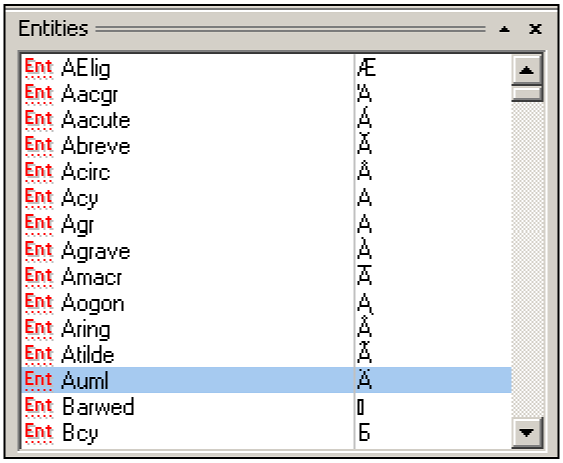
arch	
condition	
conformance	
id	
lang	
os	
remap	
revision	
revisionflag	
role	
security	
userlevel	
vendor	
xreflabel	

To enter a value for an attribute, click in the value field of the attribute and enter the value. This creates the attribute and its value in the XML document.

Note: Entering a value for an attribute will have an effect in the Authentic View display only if such an effect has been specified in the StyleVision Power Stylesheet.

Entities Entry Helper
The Entities Entry Helper allows you to insert an entity in your document. Entities can be used to insert special characters or text fragments that occur often in a document (such as the name of a company).

To insert an entity, place the cursor at the point in the text where you want to have the entity inserted, then double-click the entity in the Entities Entry Helper.



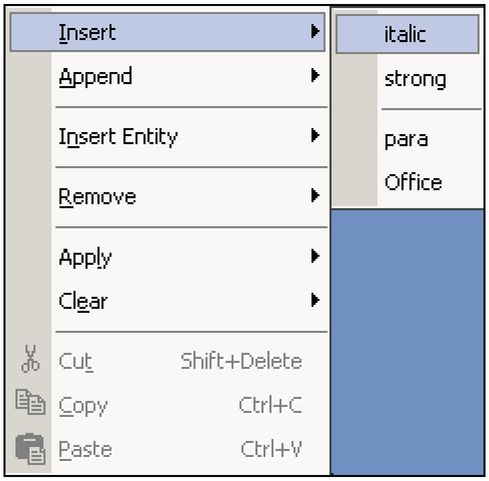
Ent AElig	Æ
Ent Aacgr	Ά
Ent Aacute	Á
Ent Abreve	Ă
Ent Acirc	Â
Ent Acy	А
Ent Agr	Α
Ent Agrave	À
Ent Amacr	Ā
Ent Aogon	Ą
Ent Aring	Å
Ent Atilde	Ã
Ent Auml	Ä
Ent Barwed	▯
Ent Bcy	Б

Note: An internal entity is one that has its value defined within the DTD. An external entity is one that has its value contained in an external source, e.g. another XML file. Both internal and external entities are listed in the Entities Entry Helper. When you insert an entity, whether internal or external, the entity—not its value—is inserted into the XML text. If the entity is an internal entity, Authentic View displays **the value of the entity**. If the entity is an external entity, Authentic View displays the entity—and not its value. This means, for example, that an XML file that is an external entity will be shown in the Authentic View display as an entity; its content does not replace the entity in the Authentic View display.

You can also **define your own entities** in Authentic View: see Define Entities in the How To Use section.

6.5 Authentic View context menus

Right-clicking on some selected document content or node pops up a menu with commands relevant to the selection or cursor location. This kind of menu is called a context menu. The context menu that appears depends on where in the document the cursor is. The context menu, with applicable commands enabled, is shown below. The figure below also shows the Insert submenu, which is a list of all elements that can be inserted at that point. Note that the **Insert submenu** shows (above the rule) the elements that can be **inserted within the current element**, and (below the rule) the elements that can be **inserted before the current element**. In the figure below, the current element is the `para` element. The `italic` and `strong` elements can be inserted within the current `para` element. The `para` and `Office` elements can be inserted before the current `para` element.



Most of the commands available in the context menu are explained in Authentic View entry helpers.

Remove element
Pointing over the **Remove** command pops up a menu list consisting of the selected element and all its ancestors up to the document element. Click the element to be removed. This is a quick way to delete an element or any of its ancestors. Note that clicking an ancestor element will remove all its descendants, including the selected element.

6.6 Features and their usage

This section describes important features of Authentic View in detail. Features have been included in this section either because they are commonly used or require an explanation of the mechanisms or concepts involved.

The section explains the following:

- There are three distinct types of tables used in Authentic View. The section Using tables in Authentic View explains the three types of tables (static SPS, dynamic SPS, and XML), and when and how to use them. It starts with the broad, conceptual picture and moves to the details of usage.
- The Date Picker is a graphical calendar that enters dates in the correct XML format when you click a date. See Using the Date Picker.
- An entity is shorthand for a special character or text string. You can define your own entities, which allows you to insert these special characters or text strings by inserting the corresponding entities. See Define Entities for details.
- What image formats can be displayed in Authentic View.

6.6.1 Tables in Authentic View

The three table types fall into two categories: SPS tables (static and dynamic) and XML tables.

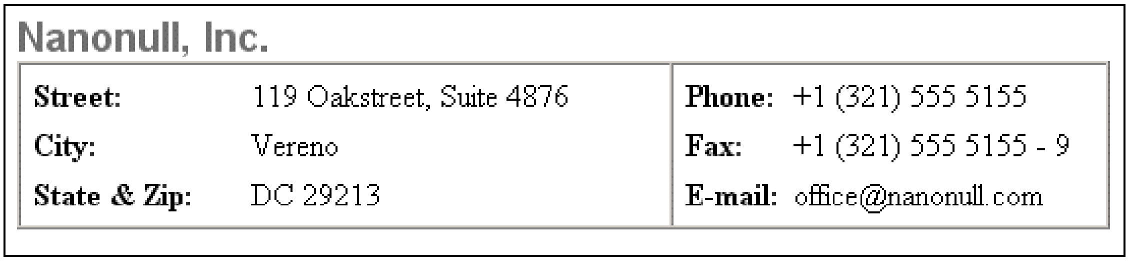
SPS tables are of two types: static and dynamic. SPS tables are designed by the designer of the StyleVision Power Stylesheet to which your XML document is linked. You yourself cannot insert an SPS table into the XML document, but you can enter data into SPS table fields and add and delete the rows of dynamic SPS tables. The section on SPS tables below explains the features of these tables.

XML tables are inserted by you, the user of Authentic View. Their purpose is to enable you to insert tables at any allowed location in the document hierarchy should you wish to do so. The editing features of XML tables and the XML table editing icons are described below.

SPS Tables

Two types of SPS tables are used in Authentic View: static tables and dynamic tables.

Static tables are fixed in their structure and in the content-type of cells. You, as the user of Authentic View, can enter data into the table cells but you cannot change the structure of these tables (i.e. add rows or columns, etc) or change the content-type of a cell. You enter data either by typing in text, or by selecting from options presented in the form of check-box or radio button alternatives or as a list in a combo-box. After you enter data, you can edit it.

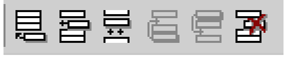


Nanonull, Inc.

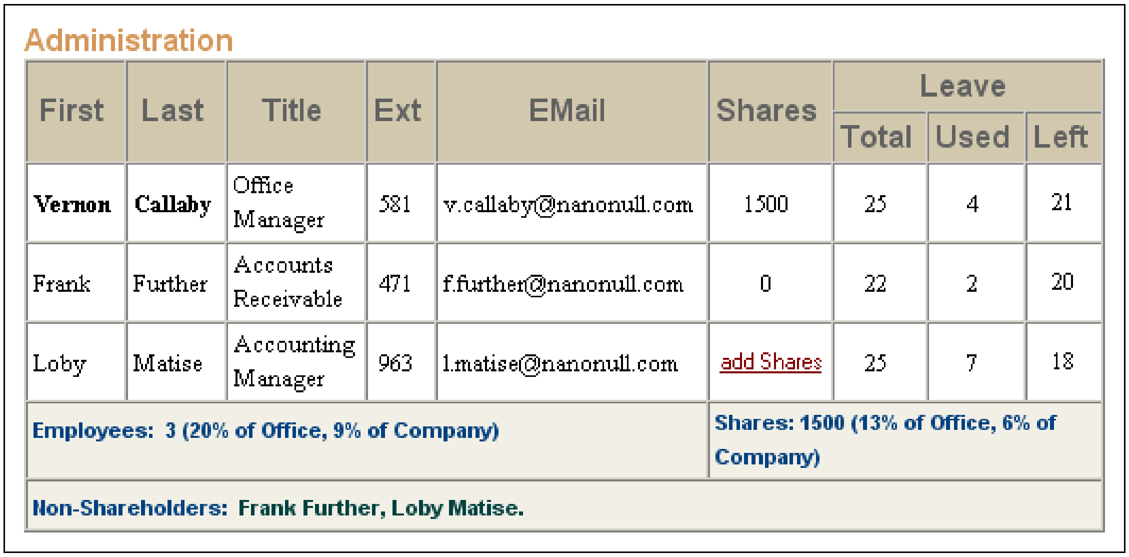
Street:	119 Oakstreet, Suite 4876	**Phone:**	+1 (321) 555 5155
City:	Vereno	**Fax:**	+1 (321) 555 5155 - 9
State & Zip:	DC 29213	**E-mail:**	office@nanonull.com

Note: The icons or commands for editing dynamic tables **must not** be used to edit static tables.

Dynamic tables have rows that represent a repeating data structure, i.e. each row has an identical data structure (not the case with static tables). Therefore, you can perform row operations: append row, insert row, move row up, move row down, and delete row. These commands are available under the **Authentic** menu and as icons in the toolbar (shown below).



To use these commands, place the cursor anywhere in the appropriate row, and then select the required command.



Administration

First	Last	Title	Ext	EMail	Shares	Leave		
						Total	Used	Left
Vernon	**Callaby**	Office Manager	581	v.callaby@nanonull.com	1500	25	4	21
Frank	Further	Accounts Receivable	471	f.further@nanonull.com	0	22	2	20
Loby	Matise	Accounting Manager	963	l.matise@nanonull.com	add Shares	25	7	18
Employees: 3 (20% of Office, 9% of Company)					**Shares: 1500 (13% of Office, 6% of Company)**			
Non-Shareholders: Frank Further, Loby Matise.								

To move among cells in the table, use the Up, Down, Left, and Right arrow keys. To move forward from one cell to the next, use the Tab key. Pressing the Tab key in the last cell of a row creates a new row.

XML Tables

XML tables can be inserted by you, the user of Authentic View. They enable you to insert tables anywhere in the XML document where they are allowed, which is useful if you need to insert tabular information in your document. These tables will be printed out as tables when you print out directly from Authentic View. If you are also generating output with XSLT stylesheets, discuss the required output with the designer of the StyleVision Power Stylesheet.

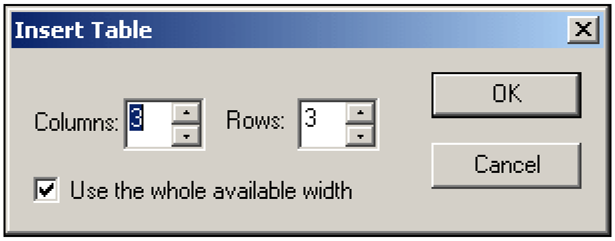
Note that you can insert XML tables only at allowed locations. These locations are specified in the schema (DTD or XML Schema). If you wish to insert a table at additional locations, discuss this with the person designing the StyleVision Power Stylesheet.

Working with XML tables

There are three steps involved when working with XML tables: inserting the table; formatting it; and entering data. The commands for working with XML tables are available as icons in the toolbar (see XML table editing icons).

Inserting tables

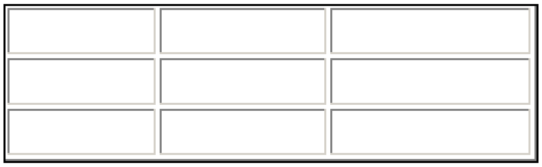
To insert an XML table place your cursor where you wish to insert the table, and click the ▦ icon. (Note that where you can insert tables is determined by the schema.) This pops up the Insert Table dialog (shown below).



Columns: 3 Rows: 3

☑ Use the whole available width

Select the number of columns and rows, and specify whether you wish the table to extend the entire available width. For the specifications given in the dialog box shown above, the following table is created.

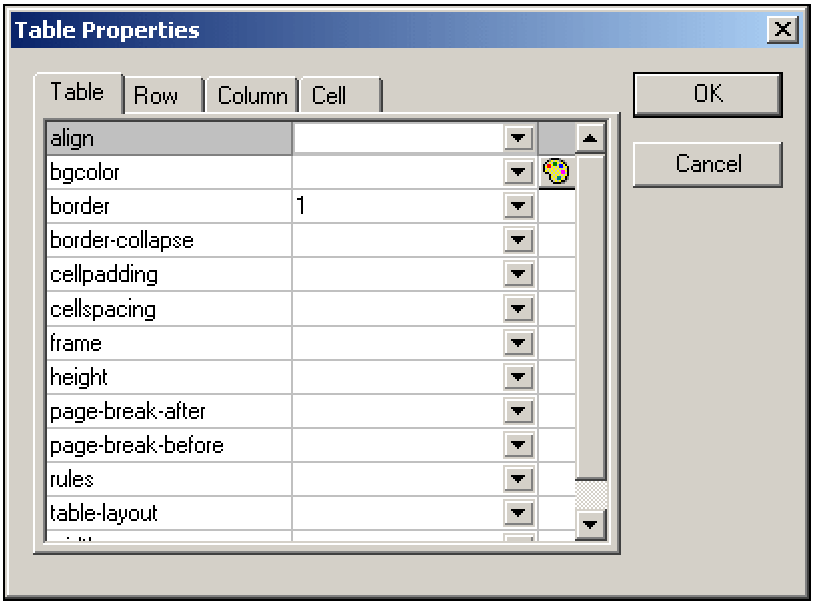


You can add and delete columns, create row and column joins later. So create the broad structure first.

Note: All modifications to table structure must be made by using the Table menu commands. They cannot be made by changing attribute values in the Attribute Entry Helper.

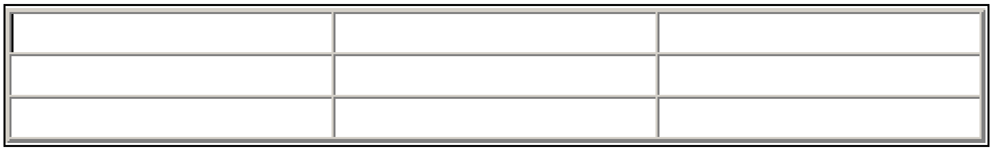
Formatting tables and entering data

To format your table, place the cursor anywhere in the table and click the (Table Properties) icon. This pops up the Table Properties dialog (shown below), where you specify formatting for the table, or for a row, column, or cell.

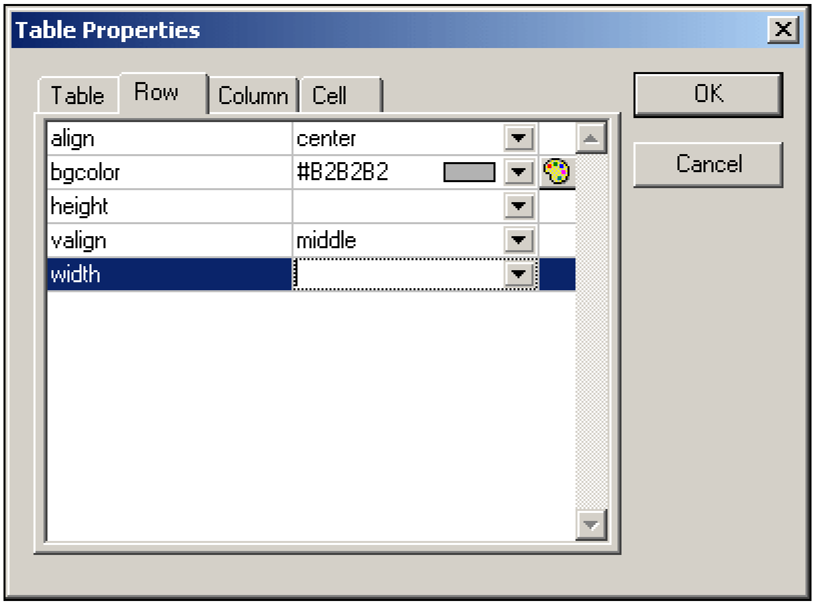


align	
bgcolor	
border	1
border-collapse	
cellpadding	
cellspacing	
frame	
height	
page-break-after	
page-break-before	
rules	
table-layout	

Set the cellspacing and cellpadding properties to "0". Your table will now look like this:

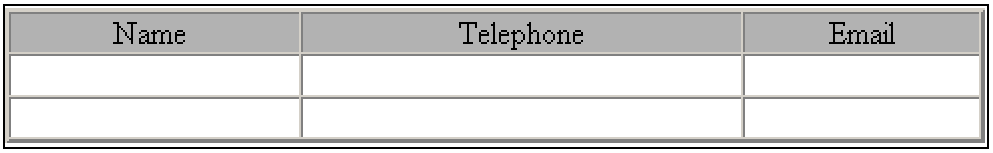


Now place the cursor in the first row to format it, and click the (Table Properties) icon. Click the Row tab.



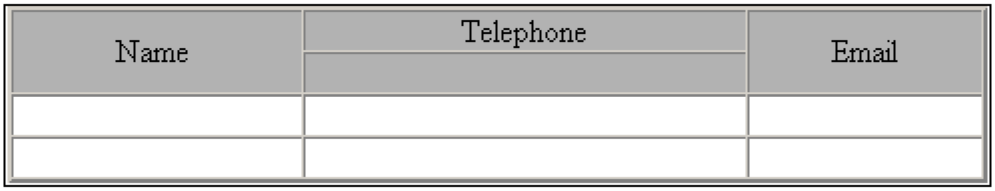
align	center
bgcolor	#B2B2B2
height	
valign	middle
width	

Since the first row will be the header row, set a background color to differentiate this row from the other rows. Note the Row properties that have been set in the figure above. Then enter the column header text. Your table will now look like this:



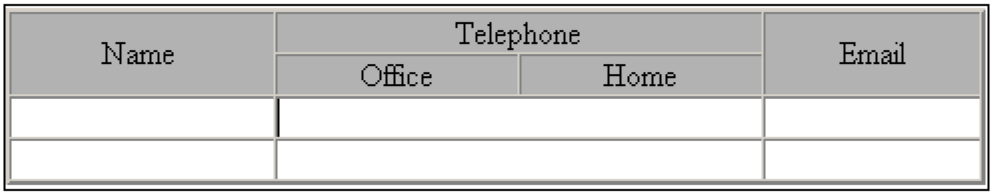
Name	Telephone	Email

Notice that the alignment is centered as specified. Now, say you want to divide the "Telephone" column into the sub-columns "Office" and "Home", you would need to join cells. Place the cursor in the "Telephone" cell, and click the (Split vertically) icon. Your table will look like this:



Name	Telephone	Email

Now place the cursor in the cell below the cell containing "Telephone", and click the (Split horizontally) icon. Then type in the column headers "Office" and "Home". Your table will now look like this:



Name	Telephone		Email
	Office	Home	

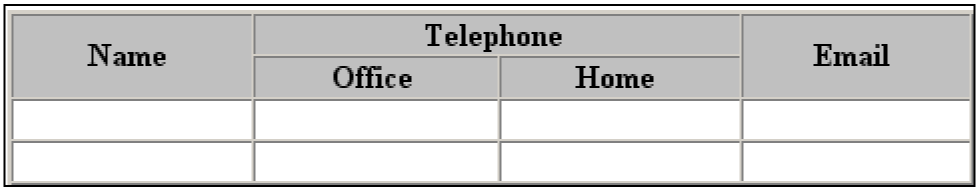
Now you will have to vertically split each cell in the "Telephone" column.

You can also add and delete columns and rows, and vertically align cell content, using the table-editing icons. The XML table editing icons are described in the User Reference, in the section titled "XML Table Icons".

Moving among cells in the table
To move among cells in the XML table, use the Up, Down, Right, and Left arrow keys.

Entering data in a cell
To enter data in a cell, place the cursor in the cell, and type in the data.

Formatting text
Text in an XML table, as with other text in the XML document, must be formatted using XML elements or attributes. To add an element, highlight the text and double-click the required element in the Elements Entry Helper. To specify an attribute value, place the cursor within the text fragment and enter the required attribute value in the Attributes Entry Helper. After formatting the header text bold, your table will look like this.



Name	**Telephone**		**Email**
	Office	**Home**	

The text above was formatted by highlighting the text, and double-clicking the element `strong`, for which a global template exists that specifies bold as the font-weight. The text formatting becomes immediately visible.

Note: For text formatting to be displayed in Authentic View, a global template with the required text formatting must have been created in StyleVision for the element in question.

XML Table Editing Icons

The commands required to edit XML tables are available as icons in the toolbar, and are listed below. Note that no corresponding menu commands exist for these icons.

For a full description of when and how XML tables are to be used, see XML tables.

Insert table

The "Insert Table" command inserts a **CALS / HTML table** at the current cursor position.

Delete table

The "Delete table" command deletes the currently active table.

Append row

The "Append row" command appends a row to the end of the currently active table.

Append column

The "Append column" command appends a column to the end of the currently active table.

Insert row

The "Insert row" command inserts a row above the current cursor position in the currently active table.

Insert column

The "Insert column" command inserts a column to the left of the current cursor position in the currently active table.

Join cell left

The "Join cell left" command joins the current cell (current cursor position) with the cell to the left. The tags of both cells remain in the new cell, the column headers remain unchanged.

Join cell right

The "Join cell right" command joins the current cell (current cursor position) with the cell to the right. The tags of both cells remain in the new cell, the column headers remain unchanged.

Join cell below

The "Join cell below" command joins the current cell (current cursor position) with the cell below. The tags of both cells remain in the new cell, the column headers remain unchanged.

Join cell above

The "Join cell above" command joins the current cell (current cursor position) with the cell above. The tags of both cells remain in the new cell, the column headers remain unchanged.

Split cell horizontally

The "Split cell Horizontally" command creates a new cell to the right of the currently active cell. The size of both cells, is now the same as the original cell.

Split cell vertically

The "Split cell Vertically" command creates a new cell below the currently active cell.

Align top

This command aligns the cell contents to the top of the cell.

Center vertically

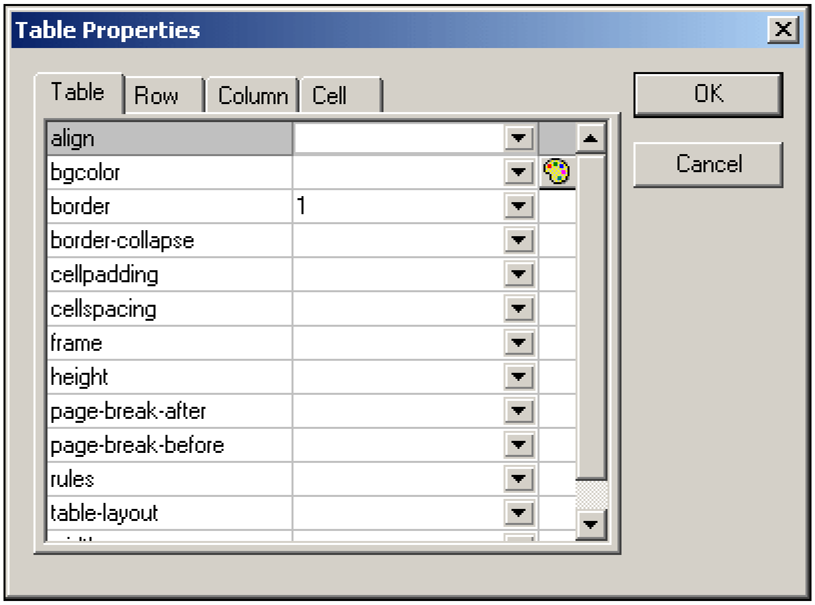
This command centers the cell contents.

Align bottom

This command aligns the cell contents to the bottom of the cell.

Table properties

The "Table properties" command opens the Table Properties dialog box. This icon is only made active for HTML tables, it cannot be clicked for CALS tables.



align	
bgcolor	
border	1
border-collapse	
cellpadding	
cellspacing	
frame	
height	
page-break-after	
page-break-before	
rules	
table-layout	

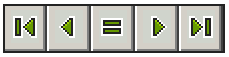
6.6.2 Editing a DB

In Authentic View, you can edit database (DB) tables and save data back to a DB. This section contains a full description of interface features available to you when editing a DB table. The following general points need to be noted:

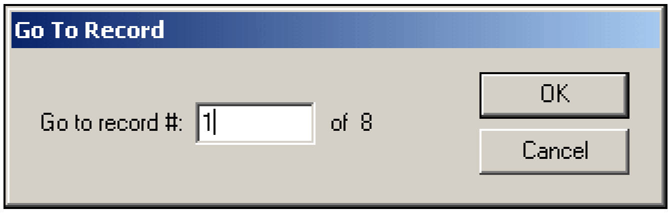
- The number of records in a DB table that are displayed in Authentic View may have been deliberately restricted by the designer of the StyleVision Power Stylesheet in order to make the design more compact. In such cases, only that limited number of records is initially loaded into Authentic View. Using the DB table row navigation icons (*see* Navigating a DB Table), you can load and display the other records in the DB table.
- You can query the DB to display certain records.
- You can add, modify, and delete DB records, and save your changes back to the DB. See Modifying a DB Table.

Navigating a DB Table

The commands to navigate DB table rows are available as buttons in the Authentic View document. Typically, one navigation panel with either four or five buttons accompanies each DB table.



The arrow icons are, from left to right, Go to First Record in the DB; Go to Previous Record; Open the Go to Record # dialog (*shown below*); Go to Next Record; and Go to Last Record.



Go To Record

Go to record #: 1 of 8

To navigate a DB table, click the required button.

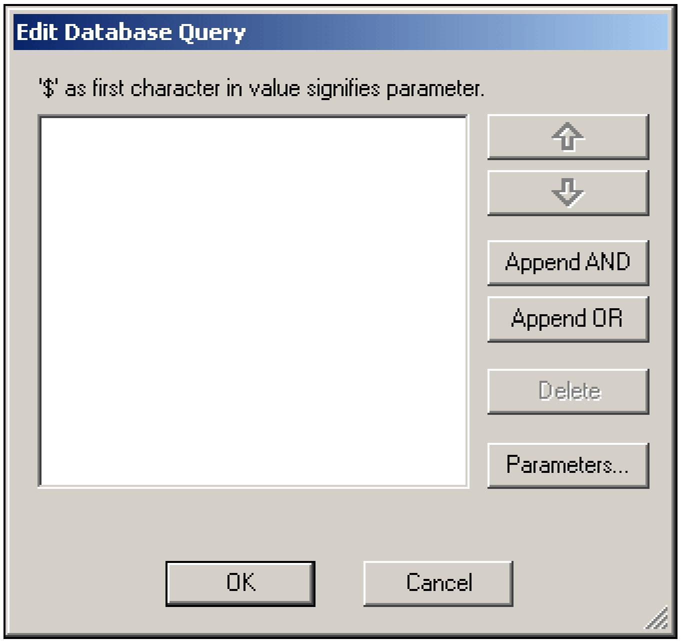
DB Queries

A DB query enables you to query the records of a table displayed in Authentic View. A query is made for an individual table, and only one query can be made for each table. You can make a query at any time while editing. If you have unsaved changes in your Authentic View document at the time you submit the query, you will be prompted about whether you wish to save **all** changes made in the document or discard **all** changes. Note that even changes made in other tables will be saved/discarded. After you submit the query, the table is reloaded using the query conditions.

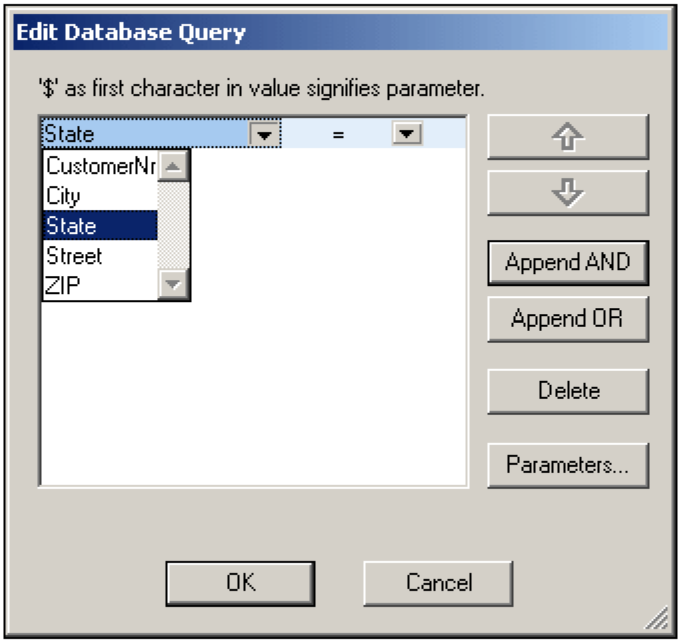
Note: If you get a message saying that too many tables are open, then you can reduce the number of tables that are open by using a query to filter out some tables.

To create and submit a query, do the following:

1. Click the Query button for the required table in order to pop up the Edit Database Query dialog (*shown below*). This button typically appears at the top of each DB table or below it. If a Query button is not present for any table, the designer of the StyleVision Power Stylesheet has not enabled the DB Query feature for that table.



2. Click the **Append AND** or **Append OR** button. This appends an empty criterion for the query (shown below).



4. Enter the expression for the criterion. An expression consists of: (i) a field name (available from the associated combo-box); (ii) an operator (available from the associated combo-box); and (iii) a value (to be entered directly). For details of how to construct expressions see the Expressions in criteria section below.
5. If you wish to add another criterion, click the **Append AND** or **Append OR** button according to which logical operator (AND or OR) you wish to use to join the two criteria. Then add the new criterion. For details about the logical operators, see the section Re-ordering criteria in DB Queries.

Expressions in criteria

Expressions in DB Query criteria consist of a field name, an operator, and a value. The **available field names** are the child elements of the selected top-level data table; the names of these fields are listed in a combo-box (*see screenshot above*). The **operators** you can use are listed below:

`=`	`Equal to`
`<>`	`Not equal to`
`<`	`Less than`
`<=`	`Less than or equal to`
`>`	`Greater than`
`>=`	`Greater than or equal to`
`LIKE`	`Phonetically alike`
`NOT LIKE`	`Phonetically not alike`
`IS NULL`	`Is empty`
`NOT NULL`	`Is not empty`

If IS NULL or NOT NULL is selected, the Value field is disabled. **Values** must be entered without quotes (or any other delimiter). Values must also have the same formatting as that of the corresponding DB field; otherwise the expression will evaluate to `FALSE`. For example, if a criteria for a field of the `date` datatype in an MS Access DB has an expression `StartDate=25/05/2004`, the expression will evaluate to `FALSE` because the `date` datatype in an MS Access DB has a format of `YYYY-MM-DD`.

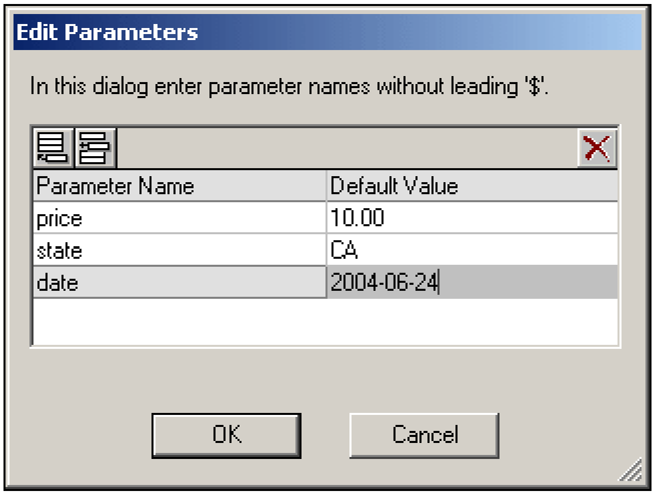
Using parameters with DB Queries
You can also enter the name of a parameter as the value of an expression. This causes the parameter to be called and its value to be used as the value of that expression. The parameter you enter here can be a parameter that has already been declared for the stylesheet, or it can be a parameter that you declare subsequently to using it in an expression.

Parameters are useful if you wish to use a single value in multiple expressions.

To enter the name of a parameter as the value of an expression, type `$` into the value input field followed (without any intervening space) by the name of the parameter. If the parameter has already been declared, then the entry will be colored green. If the parameter has not been declared, the entry will be red, and you must declare it.

Declaring parameters from the Edit DB Query dialog
To access the Edit Parameters dialog (in order to declare parameters), do the following:

1. Click the **Parameters...** button in the Edit Database Query dialog. This pops up the **Edit Parameters** dialog shown below.



Edit Parameters

In this dialog enter parameter names without leading '$'.

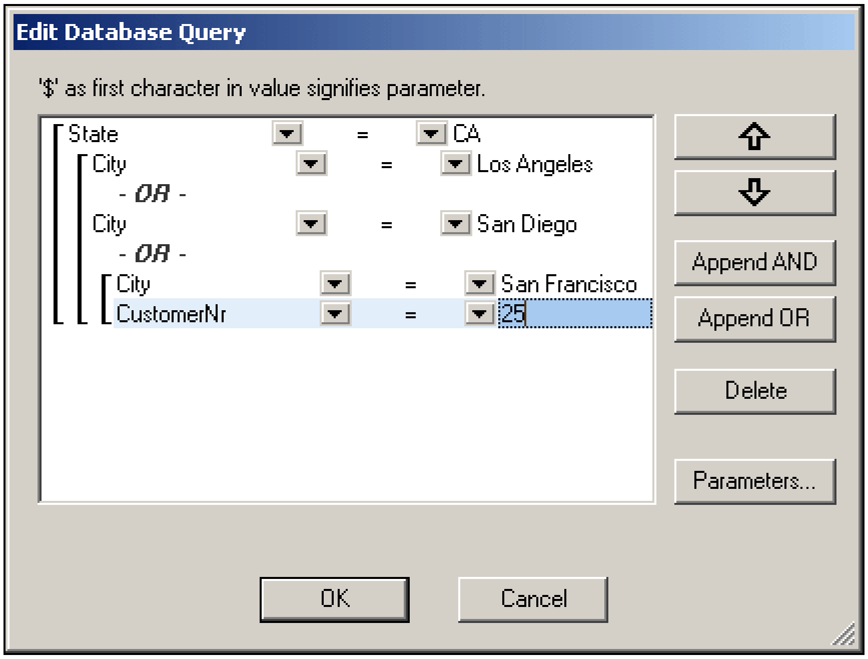
Parameter Name	Default Value
price	10.00
state	CA
date	2004-06-24

2. Type in the name and value of the parameter in the appropriate fields.

Note: The Edit Parameters dialog contains **all** the parameters that have been defined for the stylesheet. While it is an error to use an undeclared parameter in the StyleVision Power Stylesheet, it is not an error to declare a parameter and not use it.

Re-ordering criteria in DB Queries

The logical structure of the DB Query and the relationship between any two criteria or sets of criteria is indicated graphically. Each level of the logical structure is indicated by a square bracket. Two adjacent criteria or sets of criteria indicate the AND operator, whereas if two criteria are separated by the word OR then the OR operator is indicated. The criteria are also appropriately indented to provide a clear overview of the logical structure of the DB Query.



Edit Database Query

'$' as first character in value signifies parameter.

```
State          =  CA
  City         =  Los Angeles
    - OR -
  City         =  San Diego
    - OR -
    City       =  San Francisco
    CustomerNr =  25
```

The DB Query shown in the screenshot above may be represented in text as:

```
State=CA AND (City=Los Angeles OR City=San Diego OR (City=San
Francisco AND CustomerNr=25))
```

You can re-order the DB Query by moving a criterion or set of criteria up or down relative to the other criteria in the DB Query. To move a criterion or set of criteria, do the following:

- Select the criterion by clicking on it, or select an entire level by clicking on the bracket that represents that level.
- Click the Up or Down arrow button in the dialog.

The following points should be noted:

- If the adjacent criterion in the direction of movement is at the same level, the two criteria exchange places.
- A set of criteria (i.e. criterion within a bracket) changes position within the same level; it does not change levels.
- An individual criterion changes position within the same level. If the adjacent criterion is further outward/inward (i.e. not on the same level), then the selected criterion will move outward/inward, **one level at a time**.

To delete a criterion in a DB Query, select the criterion and click **Delete**.

Modifying a DB Query

To modify a DB Query, click the Query button . This pops up the Edit Database Query dialog box. You can now edit the expressions in any of the listed criteria, add new criteria, re-order criteria, or delete criteria in the DB Query. After you have completed the modifications, click OK. The data from the DB is automatically re-loaded into StyleVision so as to reflect the modifications to the DB Query.

Modifying a DB Table

Adding a record

To add a record to a DB table, place the cursor in the DB table row and click the icon (to append a row) or the icon (to insert a row). This creates a new record in the temporary XML file. (The new record is not created in the DB till you click the **File | Save Authentic XML Data...** command.) In Authentic View a row for the new record is appended to the DB table display. The `AltovaRowStatus` for this record is set to `A` (for Added).

When you enter data for the new record it is entered in bold and is underlined. This enables you to differentiate added records from existing records—if existing records have not been formatted with these text formatting properties. Datatype errors are flagged by being displayed in red.

The new record is added to the DB when you click **File | Save Authentic XML Data...**. After a new record is saved to the DB, its `AltovaRowStatus` field is initialized (indicated with `---`) and the record is displayed in Authentic View as a regular record.

Modifying a record

To modify a record, place the cursor at the required point in the DB table and edit the record as required. If the number of displayed records is limited, you may need to navigate to the required record (using the navigation icons described above).

When you modify a record, entries in all fields of the record are underlined and the `AltovaRowStatus` of all primary instances of this record is set to `U` (for Updated). All secondary instances of this record have their `AltovaRowStatus` set to `u` (lowercase). Primary and secondary instances of a record are defined by the structure of the DB—and correspondingly of the XML Schema generated from it. For example, if an Address table is included in a Customer table, then the Address table can occur in the Design Document in two types of instantiations: as the Address table itself and within instantiations of the Customer table. Whichever of these two types is modified is the type that has been primarily modified. Other types—there may be more than one other type—are secondary types. Datatype errors are flagged by being displayed in red.

The modifications are saved to the DB by clicking **File | Save Authentic XML Data...**. After a modified record is saved to the DB, its `AltovaRowStatus` field is initialized (indicated with `---`) and the record is displayed in Authentic View as a regular record.

Note:

- If even a single field of a record is modified in Authentic View, the entire record is updated when the data is saved to the DB.
- The date value `0001-01-01` is defined as a `NULL` value for some DBs, and could result in an error message.

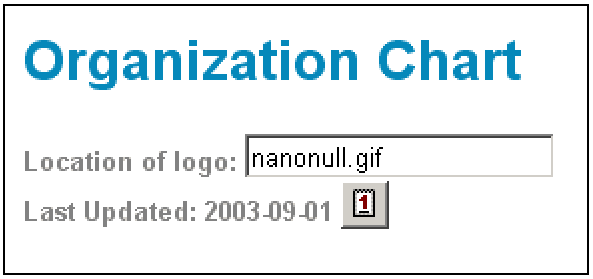
Deleting a record
To delete a record, place the cursor in the row representing the record to be deleted and click the icon. The record to be deleted is marked with a strikethrough. The `AltovaRowStatus` is set as follows: primary instances of the record are set to `D`; secondary instances to `d`; and records indirectly deleted to `X`. Indirectly deleted records are fields in the deleted record that are held in a separate table. For example, an Address table might be included in a Customer table. If a Customer record were to be deleted, then its corresponding Address record would be indirectly deleted. If an Address record in the Customer table were deleted, then the Address record in the Customer table would be primarily deleted, but the same record would be secondarily deleted in an independent Address table if this were instantiated.

The modifications are saved to the DB by clicking **File | Save Authentic XML Data...**.

Note: Saving data to the DB resets the Undo command. So you cannot undo actions that were carried out prior to the save.

6.6.3 Date Picker

The Date Picker is a graphical calendar used to enter dates in a standard format into the XML document. Having a standard format is important for the processing of data in the document. The Date Picker icon appears near the date field it modifies; it is shown below.

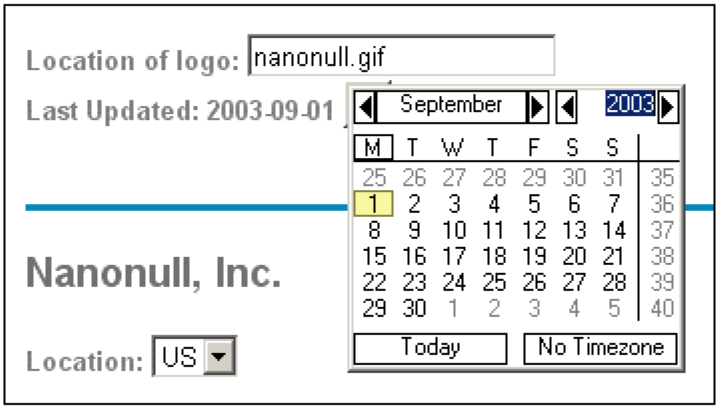


Organization Chart

Location of logo: nanonull.gif

Last Updated: 2003-09-01

To display the Date Picker (shown below), click the Date Picker icon.



Location of logo: nanonull.gif

Last Updated: 2003-09-01

Nanonull, Inc.

Location: US

September 2003

M	T	W	T	F	S	S	
25	26	27	28	29	30	31	35
1	2	3	4	5	6	7	36
8	9	10	11	12	13	14	37
15	16	17	18	19	20	21	38
22	23	24	25	26	27	28	39
29	30	1	2	3	4	5	40

To select a date, click on the desired date, month, or year. The date is entered in the XML document, and the date in the display is modified accordingly. You can also enter a time zone if this is required.

6.6.4 Define Entities

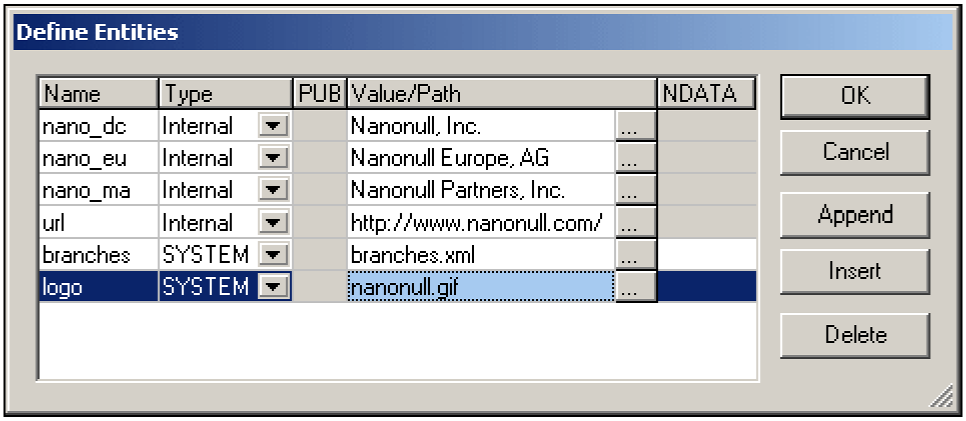
You can define entities for use in Authentic View, whether your document is based on a DTD or an XML Schema. Once defined, these entities are displayed in the Entities Entry Helper and in the **Insert Entity** submenu of the context menu. When you double-click on an entity in the Entities Entry Helper, that entity is inserted at the cursor insertion point.

An entity is useful if you will be using a text string, XML fragment, or some other external resource in multiple locations in your document. You define the entity, which is basically a short name that stands in for the required data, in the Define Entities dialog. After defining an entity you can use it at multiple locations in your document. This helps you save time and greatly enhances maintenance.

There are two broad types of entities you can use in your document: a **parsed entity**, which is XML data (either a text string or a fragment of an XML document), or an **unparsed entity**, which is non-XML data such as a binary file (usually a graphic, sound, or multimedia object). Each entity has a name and a value. In the case of parsed entities the entity is a placeholder for the XML data. The value of the entity is either the XML data itself or a URI that points to a `.xml` file that contains the XML data. In the case of unparsed entities, the value of the entity is a URI that points to the non-XML data file.

To define an entity, do the following:

1. Click **Authentic | Define Entities...**. This opens the Define Entities dialog.



Name	Type	PUB	Value/Path	NDATA
nano_dc	Internal		Nanonull, Inc.	
nano_eu	Internal		Nanonull Europe, AG	
nano_ma	Internal		Nanonull Partners, Inc.	
url	Internal		http://www.nanonull.com/	
branches	SYSTEM		branches.xml	
logo	SYSTEM		nanonull.gif	

2. Enter the name of your entity in the Name field. This is the name that will appear in the Entities Entry Helper.
3. Enter the type of entity from the drop-down list in the Type field. Three types are possible. An **Internal** entity is one for which the text to be used is stored in the XML document itself. Selecting **PUBLIC** or **SYSTEM** specifies that the resource is located outside the XML file, and will be located with the use of a public identifier or a system identifier, respectively. A system identifier is a URI that gives the location of the resource. A public identifier is a location-independent identifier, which enables some processors to identify the resource. If you specify both a public and system identifier, the public identifier resolves to the system identifier, and the system identifier is used.
4. If you have selected PUBLIC as the Type, enter the public identifier of your resource in the PUBLIC field. If you have selected Internal or SYSTEM as your Type, the PUBLIC field is disabled.
5. In the Value/Path field, you can enter any one of the following:

   - If the entity type is Internal, enter the text string you want as the value of your entity. Do not enter quotes to delimit the entry. Any quotes that you enter will be treated as part of the text string. Note that entities are a good mechanism for including Unicode characters in your document; do this by entering the Unicode number as the value of an internal entity.
   - If the entity type is SYSTEM, enter the URI of the resource or select a resource on your local network by using the Browse button. If the resource contains parsed data, it must be an XML file (i.e. it must have a `.xml` extension). Alternatively, the resource can be a binary file, such as a GIF file.
   - If the entity type is PUBLIC, you must additionally enter a system identifier in this field.

6. The NDATA entry tells the processor that this entity is not to be parsed but to be sent to the appropriate processor. The NDATA field should therefore be used with unparsed entities only.

Dialog features

You can append, insert, and delete entities by clicking the appropriate buttons. You can also sort entities on the alphabetical value of any column by clicking the column header; clicking once sorts in ascending order, twice in descending order. You can also resize the dialog box and the width of columns.

Limitations

- An entity contained within another entity is not resolved, either in the dialog, Authentic View, or XSLT output, and the ampersand character of such an entity is displayed in its escaped form, i.e. `&amp;`.
- External entities are not resolved in Authentic View, except in the case where an entity is an image file and it is entered as the value of an attribute of type `ENTITY` or

`ENTITIES`. Such entities are resolved when the document is processed with an XSLT generated from the StyleVision Power Stylesheet.

6.6.5 Images in Authentic View

Authentic View is based on Internet Explorer, and is able to display most of the image formats that your version of Internet Explorer can display. The following commonly used image formats are supported:

- GIF
- JPG
- PNG
- BMP
- WMF (Microsoft Windows Metafile)
- EMF (Enhanced Metafile)
- SVG (for PDF output only)

6.6.6 Keystrokes in Authentic View

Enter (Carriage Return) Key
In Authentic View the **Return** key is used to append additional elements when it is in certain cursor locations. For example, if the chapter of a book may (according to the schema) contain several paragraphs, then pressing **Return** inside the text of the paragraph causes a new paragraph to be appended immediately after the current paragraph. If a chapter can contain one title and several chapters, pressing Enter inside the chapter but outside any paragraph element (including within the title element) causes a new chapter to be appended after the current chapter (assuming that multiple chapters are allowed by the schema).

Note: The **Return** key does **not** insert a carriage return/line feed, i.e. it does not jump to a new line. This is the case even when the cursor is inside a text node, such as paragraph.

Browser View

7 Browser View

Browser View is typically used to view:

- XML files that have an associated XSLT file. When you switch to Browser View, the XML file is transformed on the fly using the associated XSLT stylesheet and the result is displayed directly in the browser.
- HTML files which are either created directly as HTML or created via an XSLT transformation of an XML file.

Browser View requires Microsoft's Internet Explorer 5.0 or later. If you wish to use Browser View for viewing XML files transformed by an XSLT stylesheet, we strongly recommend Internet Explorer 6.0 or later, which uses MSXML 3.0, an XML parser that fully supports the XSLT 1.0 standard. You might also wish to install MSXML 4.0. Please see our Download Center for more details.(Note that support for XSL in IE 5 is not 100% compatible with the official XSLT Recommendation. So if you encounter problems, with Browser View with IE 5, you should upgrade to IE 6.)

To view XML and HTML files in Browser View, click the Browser tab.

Browser View features

- You can open the Browser View in a separate window. To do this, switch to Browser View, and select the menu command **Browser | Separate window**. This allows you to tile windows so that you see the Browser View side-by-side with an editing view. As a result, any change you make in the editing view can be seen immediately in the Browser View: Simply press **F5** in the editing view or make the Browser View window the active window (by clicking on it).
- Browser View supports Find. In Browser View, select the menu command **Edit | Find** to find text strings.
- Browser View supports common browser commands: Back, Forward, Stop, Refresh, Font Size, and Print.

XMLSpy 2005 Home Edition

User Reference

8 User Reference

The **User Reference** section contains a complete description of all XMLSpy 2005 menu commands and explains their use in general. We've tried to make this user manual as comprehensive as possible. If, however, you have questions which are not covered in the User Reference or other parts of this documentation, please look up the FAQs and Discussion Forums on the Altova website. If you are still not able to have your problem satisfactorily addressed, please do not hesitate to contact us through the Support Center on the Altova website.

Note that in the File and Edit menus, all standard Windows commands are supported, as well as additional XML- and Internet-related commands.

8.1 File Menu

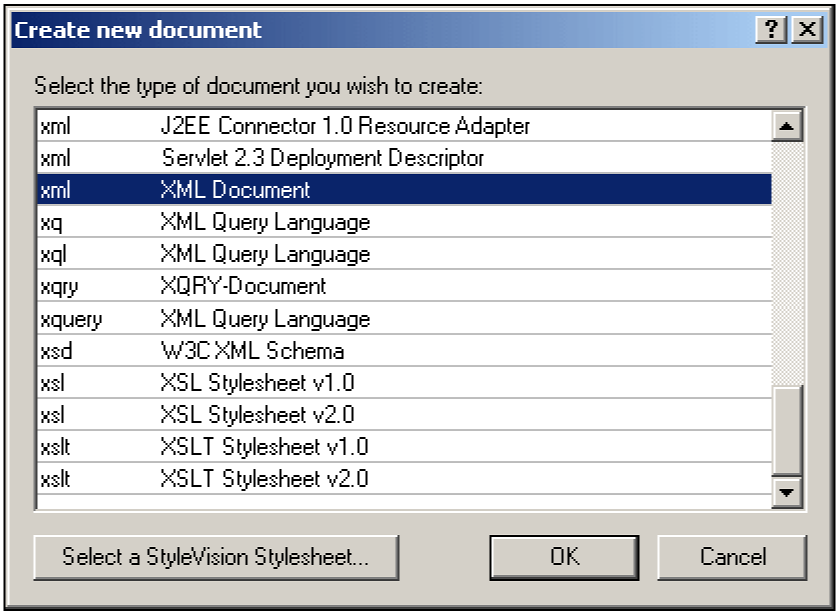
The **File** menu contains all commands relevant to manipulating files, in the order common to most Windows software products.

In addition to the standard New, Open, Save, Print, Print Setup, and Exit commands, XMLSpy 2005 offers a range of XML- and application-specific commands.

8.1.1 New...

Ctrl+N

The **New...** command is used to create a new document. Clicking **New...** pops up the Create New Document dialog, in which you can select the type of document you wish to create. If the document type you wish to create is not listed, select XML and change the file extension when you save the file. Note that you can add new file types to the list in this dialog using the Tools | Options | File types tab.

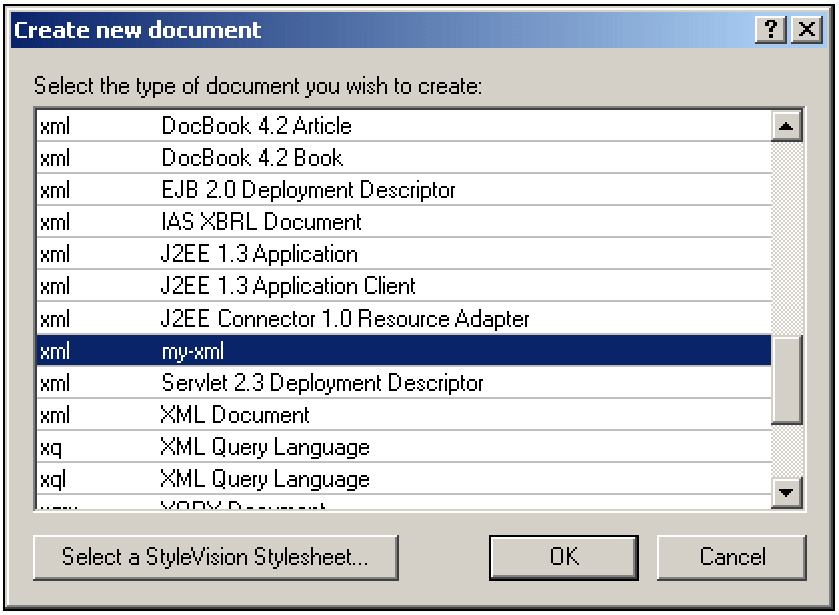

Create new document
Select the type of document you wish to create:
Select a StyleVision Stylesheet...
OK
Cancel

xml	J2EE Connector 1.0 Resource Adapter
xml	Servlet 2.3 Deployment Descriptor
xml	XML Document
xq	XML Query Language
xql	XML Query Language
xqry	XQRY-Document
xquery	XML Query Language
xsd	W3C XML Schema
xsl	XSL Stylesheet v1.0
xsl	XSL Stylesheet v2.0
xslt	XSLT Stylesheet v1.0
xslt	XSLT Stylesheet v2.0

Creating templates for new documents
You can create multiple templates for various file types. These templates can then be opened directly from the Create New Document dialog and edited. To create your own template so that it appears in the list of documents in the Create New Document dialog, you first create the template document and then save it to the folder that contains all the templates. Do the following:

1. Open the **`XMLSpy 2005\Template`** folder using Windows Explorer or your preferred navigation tool, and select a rudimentary template file from among the files named **`new.xxx`** (where `.xxx` is a file extension, such as `.xml` and `.xslt`).
2. Open the file in XMLSpy 2005, and modify the file as required. This file will be the template file.
3. When you are done, select **File | Save as...** to save the file back to the `\Template`

folder with a suitable name, say `my-xml.xml`. You now have a template called `my-xml`, which will appear in the list of files in the Create New Document dialog.



Select the type of document you wish to create:

xml	DocBook 4.2 Article
xml	DocBook 4.2 Book
xml	EJB 2.0 Deployment Descriptor
xml	IAS XBRL Document
xml	J2EE 1.3 Application
xml	J2EE 1.3 Application Client
xml	J2EE Connector 1.0 Resource Adapter
xml	my-xml
xml	Servlet 2.3 Deployment Descriptor
xml	XML Document
xq	XML Query Language
xql	XML Query Language

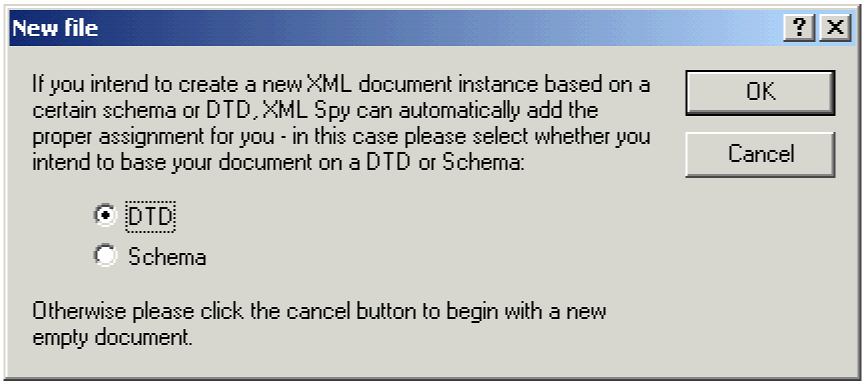
4. To open the template, select **File | New**, and then the template (`my-xml`, in this case).

Note: To delete a template, delete the template file from the template folder.

Assigning a DTD/XML Schema to a new XML document

When you create a new document of a certain type that is based on a standard schema (DTD or XML Schema), the document is automatically opened with the correct DTD or XML Schema association. For example, an XHTML file will be opened with the DTD `http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd` associated with it. And an XML Schema (`.xsd`) file is associated with the `http://www.w3.org/2001/XMLSchema` schema document.

If you are creating a new file for which the schema is not known (for example, an XML file), then you are prompted to associate a schema (DTD or XML Schema) with the document that is to be created.



If you intend to create a new XML document instance based on a certain schema or DTD, XML Spy can automatically add the proper assignment for you - in this case please select whether you intend to base your document on a DTD or Schema:

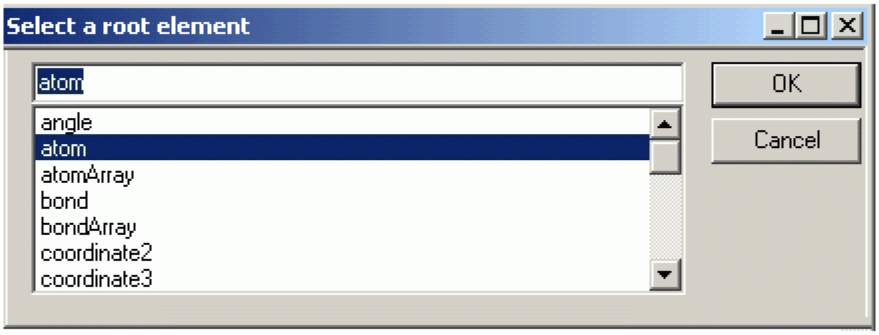
● DTD
○ Schema

Otherwise please click the cancel button to begin with a new empty document.

If you choose to associate a DTD or XML Schema with your document, clicking OK in the New File dialog enables you to browse for the schema. Clicking Cancel in this dialog will create a new file that is not associated with any schema.

Specifying the document element of a new XML document

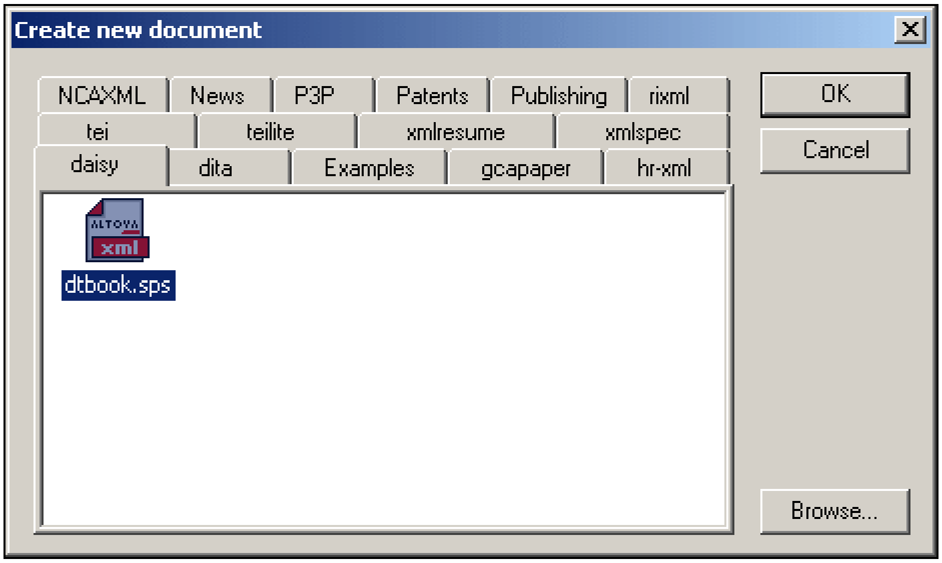
If you select an XML Schema, there can be more than one global element in it, all of which are potential document (or root) elements. You can select which of these is to be the root element of the XML document in the Select a Root Element dialog, which pops up if you select Schema in the New File dialog and if the XML Schema has more than one global element.



The new XML document is created with this element as its document element.

Assigning a StyleVision Power Stylesheet when creating a new document

When a new XML document is created, you can associate a StyleVision Power Stylesheet (`.sps` file) to view the document in Authentic View. In the Create New Document dialog (*see screenshot above*), when you click the Select StyleVision Stylesheet, the Create New Document dialog (*shown below*) appears.

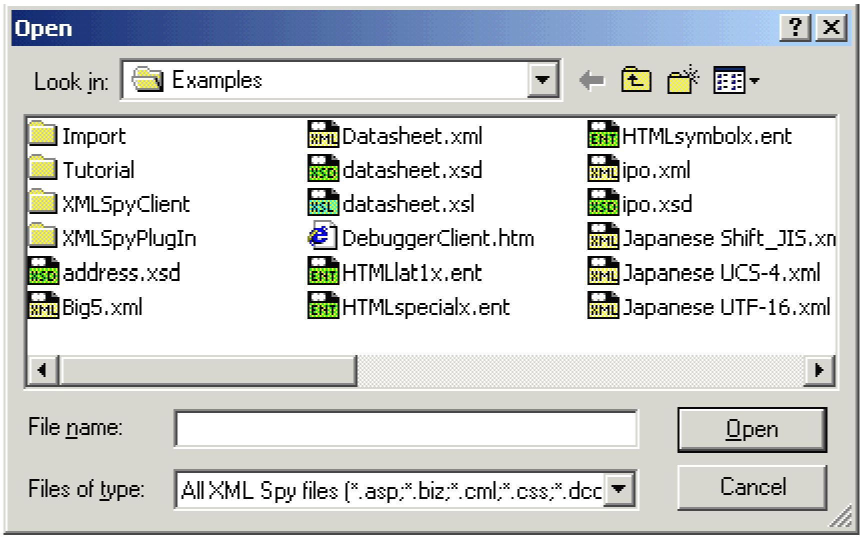


You can browse for the required StyleVision Power Stylesheet in the folder tabs displayed in the New dialog. Alternatively, you can click the **Browse...** button to navigate for and select the StyleVision Power Stylesheet. The tabs that appear in the New dialog correspond to folders in the `sps/Template` folder of your application folder.

8.1.2 Open...

Ctrl+O

The **Open...** command pops up the familiar Windows Open dialog (*screenshot below*), and allows you to open any XML-related document or text document. In the Open dialog, you can select more than one file to open. Use the Files of Type combo box to restrict the kind of files displayed in the dialog box. (The list of available file types can be configured in the File Types tab of the Options dialog (**Tools | Options**).



Look in: Examples

File name:

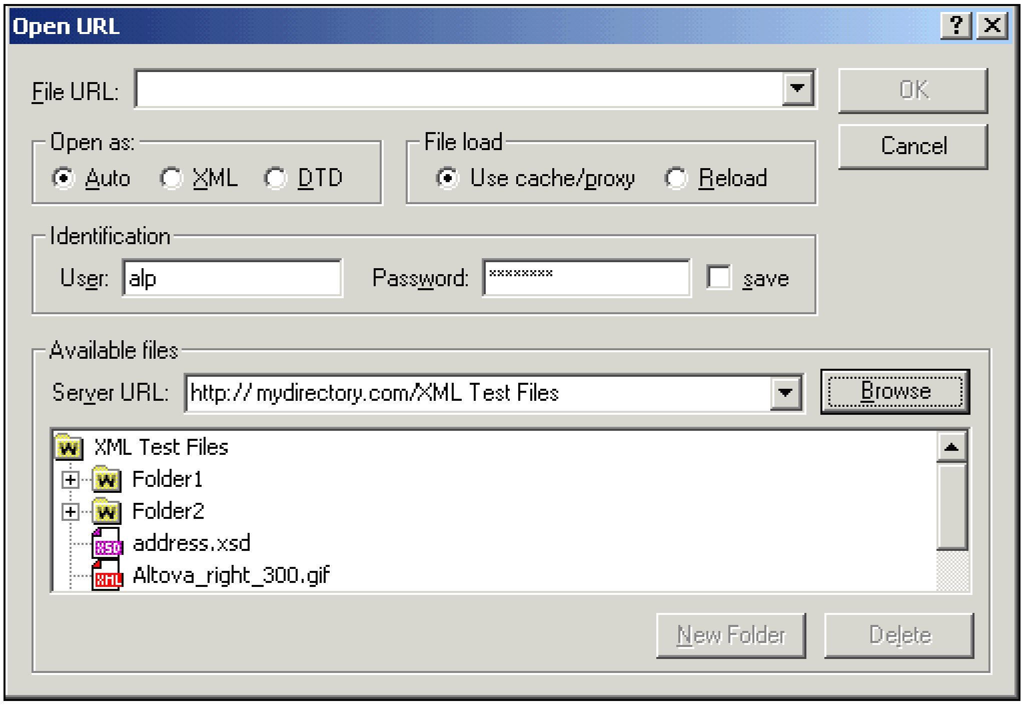
Files of type: All XML Spy files (*.asp;*.biz;*.cml;*.css;*.dcc

When an XML file is opened, it is checked for well-formedness. If the file is not well-formed, you will get a file-not-well-formed error. Fix the error and click Recheck to recheck. If you have opted for automatic validation upon opening and the file is invalid, you will get an error message. Fix the error and click Revalidate to revalidate.

8.1.3 Open URL...

The **Open URL...** command opens non-local files from a URL using http and WebDAV. To open a URL, do the following:

1. Click the Open URL command. This pops up the Open URL dialog.



Open URL

File URL:

Open as: ● Auto ○ XML ○ DTD

File load: ● Use cache/proxy ○ Reload

Identification

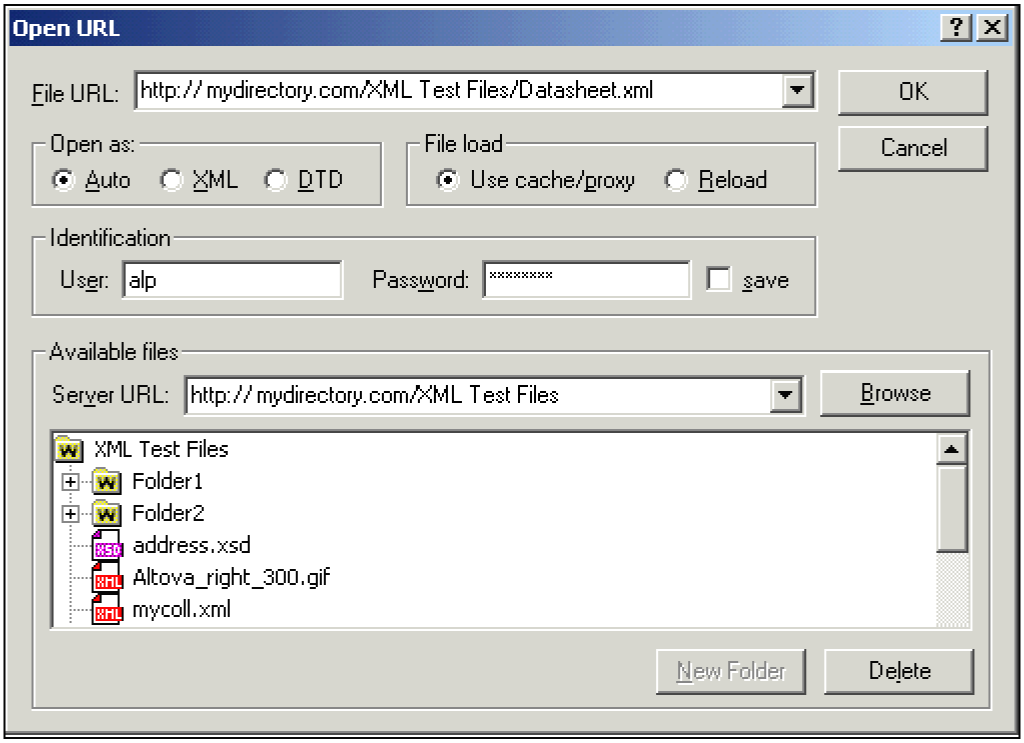
User: alp Password: ********* □ save

Available files

Server URL: http:// mydirectory.com/XML Test Files

```
XML Test Files
 + Folder1
 + Folder2
   address.xsd
   Altova_right_300.gif
```

2. Enter the URL you want to access, in the "Server URL" field.
3. Enter your User-ID in the User and Password fields, if the server is password protected.
4. Click the Browse button to view and navigate the directory structure of the server.
5. Click the file you want to load into XMLSpy 2005.



Open URL

File URL: http:// mydirectory.com/XML Test Files/Datasheet.xml

Open as: ● Auto ○ XML ○ DTD

File load: ● Use cache/proxy ○ Reload

Identification

User: alp Password: ********* □ save

Available files

Server URL: http:// mydirectory.com/XML Test Files

```
XML Test Files
 + Folder1
 + Folder2
   address.xsd
   Altova_right_300.gif
   mycoll.xml
```

The file URL appears in the File URL field. The OK button only becomes active at this point.

6. Click the OK button to load the file.

The file you open appears in the main window.

Note: The Browse function is only available on servers which support the FTP, HTTP, and HTTPS (if the server supports WebDAV) protocols, and on servers that support WebDAV.

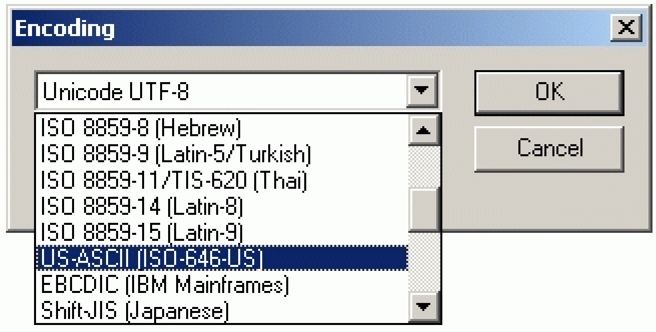
File load
To give you more control over the loading process, you can choose to load the file through the local cache or a proxy server (which considerably speeds up the process if the file has been loaded before). Alternatively, you may want to reload the file if you are working, say, with an electronic publishing or database system; select the Reload option in this case.

8.1.4 Reload

The **Reload** command allows you to reload open documents. This is useful if an open document has been modified outside XMLSpy 2005. If a modification occurs, XMLSpy 2005 asks whether you wish to reload the file. If you reload, then any changes you may have made to the file since the last save will be lost. This option can be changed in the Options dialog (Tools | Options).

8.1.5 Encoding...

The **Encoding...** command lets you view the current encoding of a file and select a different encoding when saving the current document the next time.



If you select a different encoding than the one in use before, the encoding specification in the XML declaration in the prolog will be adjusted accordingly. For 16-bit and 32-bit per character encodings (UTF-16, UCS-2, and UCS-4) you can also specify the byte-order to be used for the file. You can also enter the new encoding into the encoding specification of the XML-declaration. When saving a document, XMLSpy 2005 automatically checks the encoding specification and opens a dialog box if it cannot recognize the encoding name entered by the user.

Note: If your document contains characters that cannot be represented in the selected encoding, you will get a warning message as soon as you save your file.

8.1.6 Close

The **Close** command closes the active document window. If the file was modified (indicated by an asterisk * after the file name in the title bar), you will be asked if you wish to save the file first.

8.1.7 Close All

The **Close All** command closes all open document windows. If any document has been modified (indicated by an asterisk * after the file name in the title bar), you will be asked if you wish to save the file first.

8.1.8 Save

Ctrl+S

The **Save** command saves the contents of the active document to the file from which it has been opened. When saving a document, the file is automatically checked for well-formedness. The file will also be validated automatically if this option has been set in the File tab of the Options dialog (**Tools | Options**). The XML declaration is also checked for the encoding specification, and this encoding is applied to the document when the file is saved.

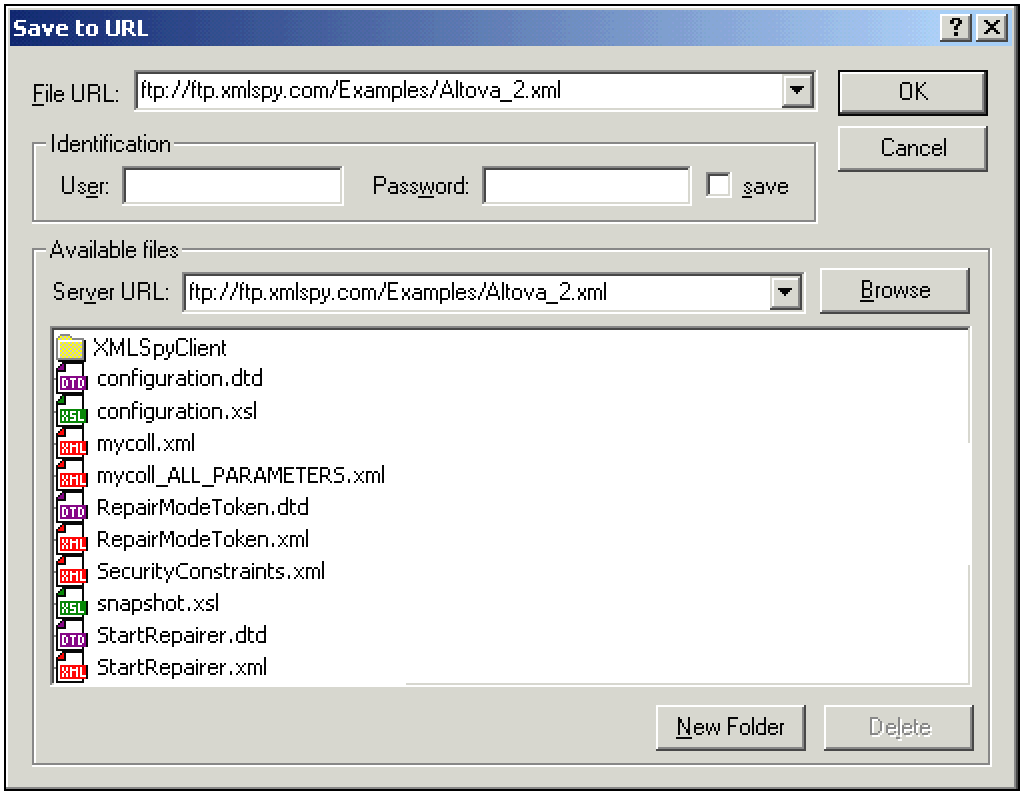
8.1.9 Save As...

The **Save As...** command pops up the familiar Windows Save As dialog box, in which you enter the name and location of the file you wish to save the active file as. The same checks and validations occur as for the **Save** command.

8.1.10 Save to URL...

The **Save to URL...** command allows you to save files to a specified URL. To save an XML document to a URL, do the following:

1. Make the file active in the Main Window.
2. Select the menu option **File | Save to URL** The following dialog appears.



File URL: ftp://ftp.xmlspy.com/Examples/Altova_2.xml

User:

Password:

□ save

Server URL: ftp://ftp.xmlspy.com/Examples/Altova_2.xml

XMLSpyClient
configuration.dtd
configuration.xsl
mycoll.xml
mycoll_ALL_PARAMETERS.xml
RepairModeToken.dtd
RepairModeToken.xml
SecurityConstraints.xml
snapshot.xsl
StartRepairer.dtd
StartRepairer.xml

3. Click the **Browse** button to see and navigate the directory structure of the server.
4. Enter the file name in the URL field or mark the file in the Available Files list box if you want to overwrite it.

Note:

- The Browse function is only available on servers which support the FTP, HTTP, and HTTPS (if the server supports WebDAV) protocols, and on servers that support WebDAV.

If the server you connect to is password protected, enter the required data and click the OK button to connect. You can also enter these data in the User and Password fields. The Save check box, supplies the password data for the logon attempt.

8.1.11 Save All

The **Save All** command saves all modifications that have been made to any open documents. The command is useful if you edit multiple documents simultaneously. If a document has not been saved before (for example, after being newly created), the Save as... dialog box is presented for that document.

8.1.12 Print...

Ctrl+P

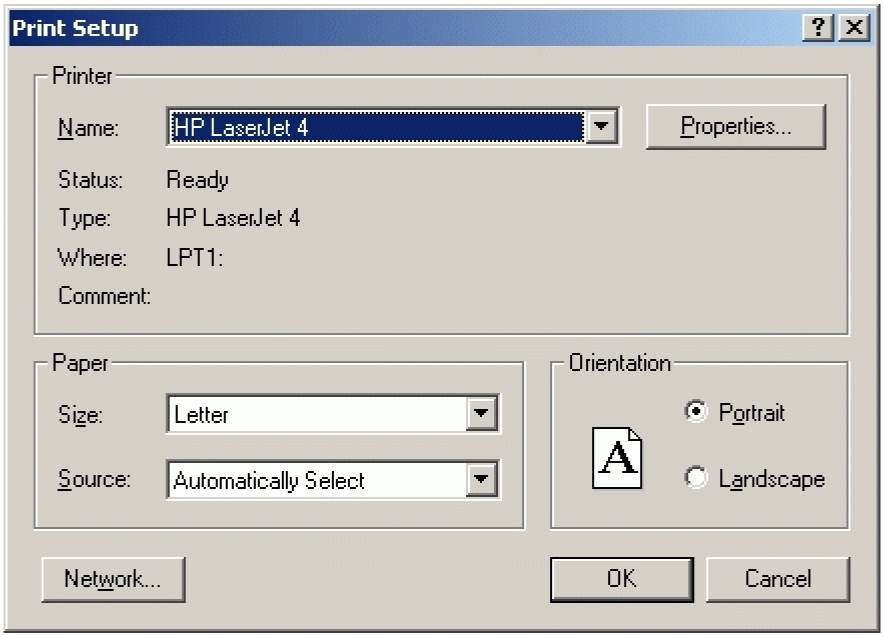
The **Print...** command opens the Print dialog box, in which you can select printer options.

8.1.13 Print Preview

The **Print Preview** command opens the Print dialog box. Click the Preview button to display a print preview of the currently active document.

8.1.14 Print Setup...

The **Print Setup...** command, displays the printer-specific Print Setup dialog box, in which you specify such printer settings as paper format and page orientation. These settings are applied to all subsequent print jobs.



Print Setup

Printer

Name: HP LaserJet 4

Status: Ready

Type: HP LaserJet 4

Where: LPT1:

Comment:

Paper

Size: Letter

Source: Automatically Select

Orientation

● Portrait

○ Landscape

The screenshot above shows the Print Setup dialog in which an HP LaserJet 4 printer attached to a parallel port (LPT1) is selected.

8.1.15 Most Recently Used Files

The **File** menu displays a list of the nine most recently used files, with the most recently opened file shown at the top of the list. You can open any of these files by clicking its name. To open a file in the list using the keyboard, press **ALT+F** to open the **File** menu, and then press the number of the file you want to open.

8.1.16 Exit

The **Exit** command is used to quit XMLSpy 2005. If you have any open files with unsaved changes, you are prompted to save these changes. XMLSpy 2005 also saves modifications to program settings and information about the most recently used files.

8.2 Edit Menu

- The **Edit** menu contains commands for editing documents in XMLSpy 2005.

8.2.1 Undo

Ctrl+ Z

The **Undo** command contains support for unlimited levels of Undo! Every action can be undone and it is possible to undo one command after another. The Undo history is retained after using the Save command, enabling you go back to the state the document was in before you saved your changes.

8.2.2 Redo

Ctrl+Y

The **Redo** command allows you to redo previously undone commands, thereby giving you a complete history of work completed. You can step back and forward through this history using the Undo and Redo commands.

8.2.3 Cut

Shift+Del or **Ctrl+X**

The **Cut** command copies the selected text or items to the clipboard and deletes them from their present location.

8.2.4 Copy

Ctrl+C

The **Copy** command copies the selected text or items to the clipboard. This can be used to duplicate data within XMLSpy 2005 or to move data to another application.

8.2.5 Paste

Ctrl+V

The **Paste** command inserts the contents of the clipboard at the current cursor position.

8.2.6 Delete

Del

The **Delete** command deletes the currently selected text or items without placing them in the clipboard.

8.2.7 Pretty-Print XML Text

The **Pretty-Print XML Text** command reformats your XML document in Text View to give a structured display of the document. Each child node is offset from its parent by the amount of space specified in the Save File option of the File tab of the Options dialog (**Tools | Options**). Note that the XML document must be well-formed for this command to work.

8.2.8 Select All

Ctrl+A

The **Select All** command selects the contents of the entire document.

8.2.9 Find...

Ctrl+F

The **Find** command pops up the Find dialog, in which you can specify the string you want to find and other options for the search. To find text, enter the text in the Find What text box or use the combo box to select from one of the last 10 search criteria, and then specify the options for the search.

8.2.10 Find next

F3

The **Find next** command repeats the last Find command to search for the next occurrence of the requested text.

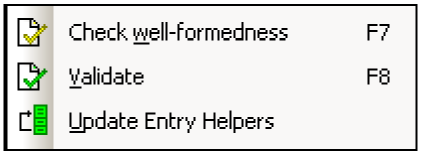
8.2.11 Replace...

Ctrl+H

The **Replace** command enables you to find an replace one text string with another text string. features the same options as the Find... command and allows you to replace the target text by any other text string of your choice. You can replace each item individually, or you can use the Replace All button to perform a global search-and-replace operation.

8.3 XML Menu

The **XML** menu contains commands commonly used when working with XML documents.



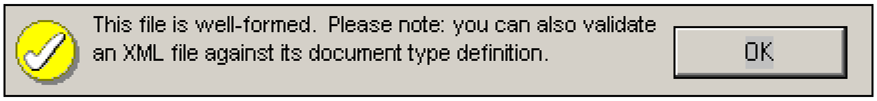
Among the most frequently used XML tasks are checks for the well-formedness of documents and validity of XML documents. Commands for these tasks are in this menu.

8.3.1 Check Well-Formedness

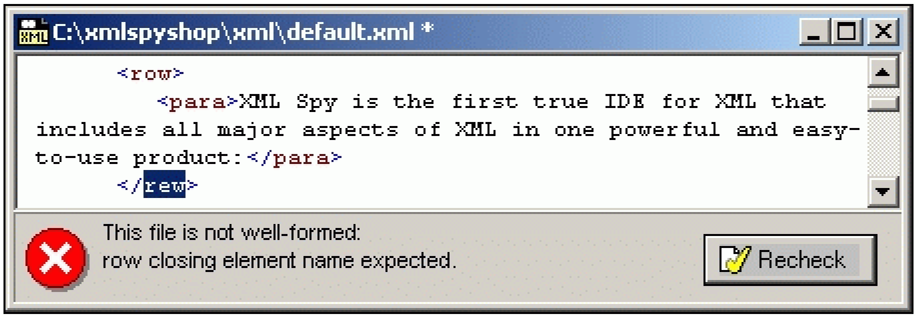
F7

Th **Check well-formedness (F7)** command checks the active document for well-formedness by the definitions of the XML 1.0 specification. Every XML document **must** be well-formed. XMLSpy 2005 checks for well-formedness whenever a document is opened or saved, or when the view is changed from Text to any other view. You can also check for well-formedness at any time while editing by using this command

If the well-formedness check succeeds when you explicitly invoke the check, a brief message is displayed at the bottom of the main window:


This file is well-formed. Please note: you can also validate an XML file against its document type definition.

If an error is encountered during the well-formedness check, the source of the problem is highlighted and a corresponding error message is shown:



```
        <row>
            <para>XML Spy is the first true IDE for XML that
includes all major aspects of XML in one powerful and easy-
to-use product:</para>
        </rew>
```

This file is not well-formed:
row closing element name expected.

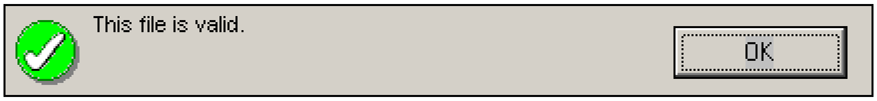
It is generally not permitted to save a malformed XML document, but XMLSpy 2005 gives you a Save Anyway option. This is useful when you want to suspend your work temporarily (in a not well-formed condition) and resume it later.

8.3.2 Validate

F8

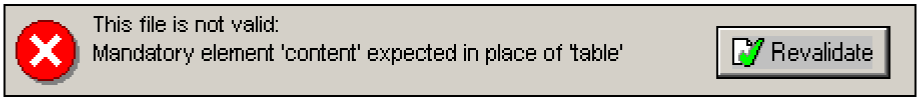
The **Validate (F8)** command enables you to validate XML documents against DTDs, XML Schemas, and other schemas. Validation is automatically carried out when you switch from Text View to any other view. You can specify that a document be automatically validated when a file is opened or saved (**Tools | Options | File**). The **Validate** command also carries out a well-formedness check before checking validity, so there is no need to use the Check Well-Formedness command before using the **Validate** command.

If a document is valid, a successful validation message is displayed at the bottom of the Main Window:



This file is valid.

Otherwise, a File Is Invalid message is displayed, which highlights the error.



This file is not valid:
Mandatory element 'content' expected in place of 'table'

Validating XML documents

To validate an XML file, make the XML document active in the Main Window, and click **XML | Validate** or **F8**. The XML document is validated against the schema referenced in the XML file. If no reference exists, an error message is displayed at the bottom of the Main Window. As long as the XML document is open, the schema is kept in memory (see Flush Memory Cache in the DTD/Schema menu).

Validating schema documents (DTDs and XML Schema)

XMLSpy 2005 supports major schema dialects, including DTD and XML Schema. To validate a schema document, make the document active in the Main Window, and click **XML | Validate** or **F8**.

XMLSpy 2005 supports a subset of the OASIS XML catalogs mechanism. These catalogs enable XMLSpy 2005 to retrieve commonly used schemas (as well as stylesheets and other files) from local user folders. This increases the overall processing speed, enables users to work offline (that is, not connected to a network), and improves the portability of documents (because URIs need be changed in the catalog files only.) The mechanism works as follows:

- A catalog file matches a `PUBLIC` system identifier to a URI that points to a local file. In XMLSpy 2005, the catalog lookup is performed by two files: `MainCatalog.xml` and `CustomCatalog.xml`. `MainCatalog.xml` maps pre-defined `PUBLIC` system identifiers of several popular schemas to URIs that point to the locally saved schema. `CustomCatalog.xml` enables you to make your own catalog extensions. Both `MainCatalog.xml` and `CustomCatalog.xml` are installed with your XMLSpy 2005 application package. The schemas, stylesheets, and other files referenced from `MainCatalog.xml` are located in specific subfolders of the your XMLSpy 2005 installation folder.
- The `PUBLIC` identifier in the `DOCTYPE` statement of your XML file will be used for the catalog lookup. For popular schemas, the `PUBLIC` identifier is usually pre-defined, thus requiring only the URI in the catalog lookup to be changed when XML documents are used on multiple machines. In XMLSpy 2005, `MainCatalog.xml` is looked up first and then `CustomCatalog.xml`.

When writing your `CustomCatalog.xml` file, use only the following subset of the OASIS catalog in order for XMLSpy 2005 to process the catalog correctly.

```
<catalog...>
<public publicId="nnn" uri="mmm"/>
<uri name="nnn" uri="mmm"/>
<system systemId="nnn" uri="mmm"/>
<rewriteURI uriIdStartString="nnn" rewritePrefix="mmm"/>
<rewriteSystem systemIdStartString="nnn" rewritePrefix="mmm"/>
```

Note:

- Although the DTDs for XML Schemas are referenced from `MainCatalog.xml`, you cannot validate your XML files against either of these schemas. The purpose of these two DTDs is to provide entry helper info for editing purposes, should you wish to create files according to these older recommendations.
- For more information on catalogs, see the `XML Catalogs specification`.

8.3.3 Update Entry-Helpers

The **Update Entry Helpers** command updates the Entry Helper windows by reloading the underlying DTD or Schema. If you have modified the XML Schema or DTD that an open XML document is based upon, it is advisable to update the Entry Helpers so that the intelligent editing information reflects the changes in the schema.

8.4 DTD/Schema Menu

The **DTD/Schema** menu contains commands that let you work efficiently with DTDs and XML Schemas.

This section contains a complete description of all the commands in this menu.

8.4.1 Assign DTD...

The **Assign DTD...** command is enabled when an XML file is active. It assigns a DTD to an XML document, thus allowing the document to be validated and enabling intelligent editing for the document. The command opens the Assign File dialog to let you specify the DTD file you wish to assign. Note that you can make the path of the assigned DTD file relative by clicking the Make Path Relative To... check box. When you are done, your XML document will contain a DOCTYPE declaration that references the assigned DTD. The DOCTYPE declaration will look something like this:

```
<!DOCTYPE main SYSTEM "http://link.xmlspy.com/spyweb.dtd">
```

Note that a DTD can be assigned to a new XML file at the time the file is created.

8.4.2 Assign Schema...

The **Assign Schema...** command is enabled when an XML document is active. It assigns an XML Schema to an XML document, thus allowing the document to be validated and enabling intelligent editing for the document. The command opens the Assign File dialog to let you specify the XML Schema file you wish to assign. Note that you can make the path of the assigned file relative by clicking the Make Path Relative To... check box. When you are done, your XML document will contain an XML Schema assignment with the required namespaces. The schema assignment will look something like this:

```
xmlns="http://www.xmlspy.com/schemas/icon/orgchart"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:schemaLocation="http://www.xmlspy.com/schemas/icon/orgchart
   http://schema.xmlspy.com/schemas/icon/orgchart.xsd"
```

The namespace declarations generated by XMLSpy 2005 depend on the kind of schema assigned and whether a target namespace has been defined in the schema document, and how it has been defined.

8.4.3 Go to DTD

The **Go to DTD** command opens the DTD on which the active XML document is based. If no DTD is assigned, then an error message is displayed.

8.4.4 Go to Schema

The **Go to Schema** command opens the XML Schema on which the active XML document is based. If no XML Schema is assigned, then an error message is displayed.

8.4.5 Go to Definition

The **Go to Definition** command displays the exact definition of an element or attribute in the corresponding Document Type Definition or Schema document.

To see the item definition in Schema/WSDL Design View

1. Use CTRL + Double click on the item you want to see the definition of, or
2. Click the item and select menu option **DTD/Schema | Go to Definition,** or click on the icon.

   In both cases, the corresponding DTD or Schema file is opened, and the item definition is highlighted.

8.4.6 Flush Memory Cache

The **Flush Memory Cache** command flushes all cached schema (DTD and XML Schema) documents from memory. To speed up validation and intelligent editing, XMLSpy 2005 caches recently used schema documents and external parsed entities in memory. Information from these cached documents is also displayed when the Go to Definition command is invoked.

Flush the memory cache if memory is tight on your system, or if you have used documents based on different schemas recently.

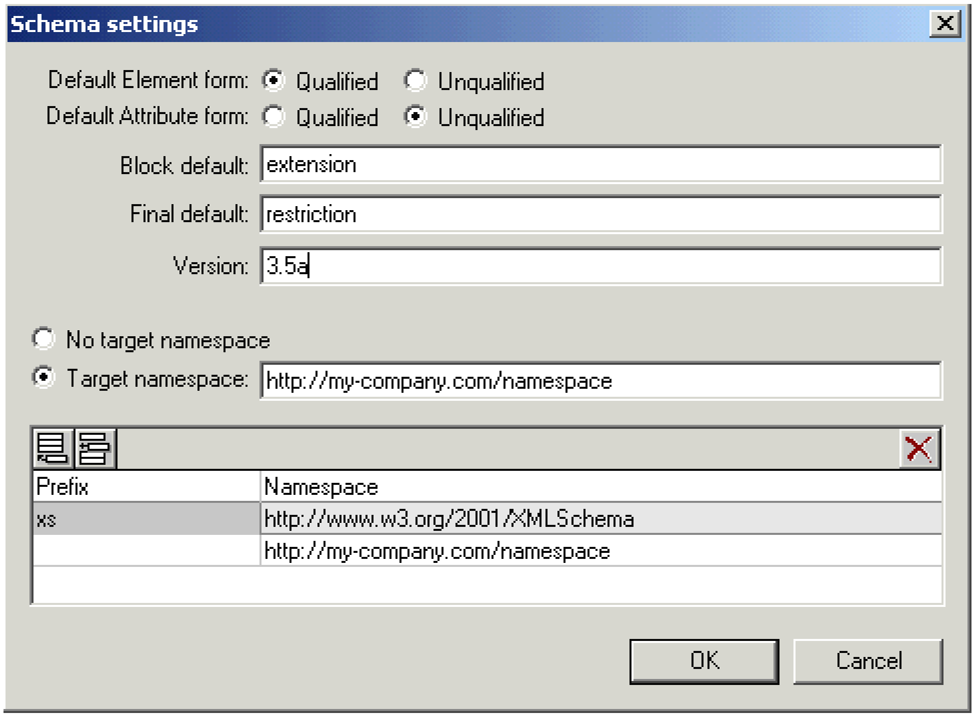
8.5 Schema Design Menu

The **Schema Design** menu enables you to design XML Schemas in a GUI. It is available when an XML Schema document is active in Schema/WSDL View.

The commands available in this menu are described in this section.

8.5.1 Schema Settings

The **Schema Settings** command lets you define global settings for the active schema. These settings are attributes and their values of the XML Schema document element, `xs:schema`.



Schema settings

Default Element form: ● Qualified ○ Unqualified

Default Attribute form: ○ Qualified ● Unqualified

Block default: extension

Final default: restriction

Version: 3.5a

○ No target namespace

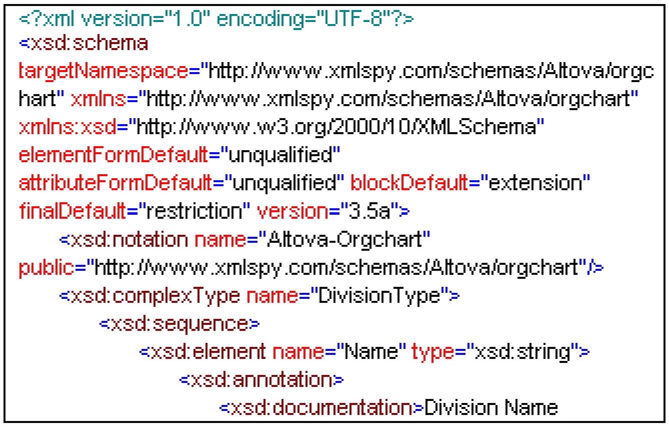
● Target namespace: http://my-company.com/namespace

Prefix	Namespace
xs	http://www.w3.org/2001/XMLSchema
	http://my-company.com/namespace

You can set the `elementFormDefault` and `attributeFormDefault` attributes of the xs:schema element each to `qualified` or `unqualified`. You can also enter the `blockDefault`, `finalDefault`, and `version` attributes by entering the required values in their respective text boxes. None of these five attributes are mandatory.

To set a target namespace for the XML instance document, select the Target Namespace radio button, and enter the target namespace in the text box. If you declare a target namespace, you must define that namespace for use in the schema document. Do this by entering a namespace line in the Namespace List in the bottom pane. You can define a prefix for this namespace, or let this namespace be the default namespace (by leaving the prefix field blank as shown in the screenshot above).

The Text View of the schema settings shown in the screenshot above will look something like this:

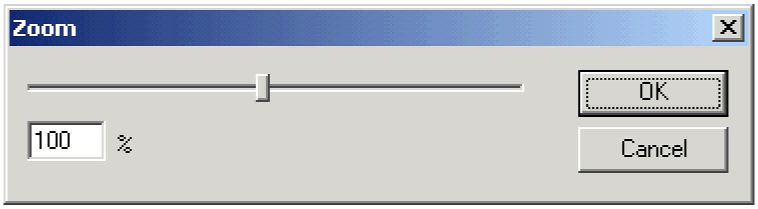


```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
targetNamespace="http://www.xmlspy.com/schemas/Altova/orgc
hart" xmlns="http://www.xmlspy.com/schemas/Altova/orgchart"
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="unqualified"
attributeFormDefault="unqualified" blockDefault="extension"
finalDefault="restriction" version="3.5a">
    <xsd:notation name="Altova-Orgchart"
public="http://www.xmlspy.com/schemas/Altova/orgchart"/>
    <xsd:complexType name="DivisionType">
        <xsd:sequence>
            <xsd:element name="Name" type="xsd:string">
                <xsd:annotation>
                    <xsd:documentation>Division Name
```

Note: These settings apply to the active schema document only.

8.5.2 Zoom

The **Zoom** command controls the zoom factor of the Content Model View. This feature is useful if you have a large content model and wish to zoom out so that the entire content model fits in the Main Window. You can zoom between 10% and 200% of actual size.

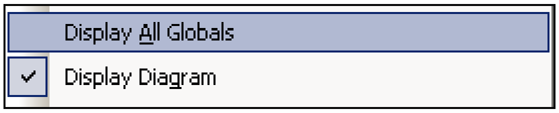


100 %

To zoom in and out, either drag the slider or click in the entry box and enter a percentage value.

8.5.3 Display All Globals

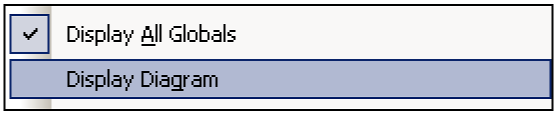
The **Display All Globals** command switches from Content Model View to Schema Overview to display all global components in the schema. It is a toggle with the Display Diagram command. The currently selected toggle is indicated with a check mark to its left (*screenshot below*).



Alternatively, you could use the **Display All Globals** icon at the top of the Content Model View to switch to the Schema Overview.

8.5.4 Display Diagram

The **Display Diagram** command switches to the Content Model View of the selected global component—if the selected component has a content model. Global components that have a content model (complex types, elements, and element groups) are indicated with the icon to its left. The Display Diagram command is a toggle with the Display All Globals command. The currently selected toggle is indicated with a check mark to its left (*screenshot below*).



Alternatively, you could use the following methods to switch to Content Model View.

- Click the icon next to the component, the content model of which you want to display.
- Double-click a component name in the Component Navigator Entry Helper (at top right).

8.6 XSL/XQuery Menu

The XSL Transformation language lets you specify how an XML document should be converted into other XML documents or text files. One kind of XML document that is generated with an XSLT document is an FO document, which can then be further processed to generate PDF output. XMLSpy 2005 contains built-in XSLT processors (for XSLT 1.0 and XSLT 2.0) and can link to an FO processor on your system to transform XML files and generate various kinds of outputs. The location of the FO processor must be specified in the XSL tab of the Options dialog (Tools | Options) in order to be able to use it directly from within the XMLSpy 2005 interface.

XMLSpy 2005 also has a built-in XQuery engine, which can be used to execute XQuery documents (with or without reference to an XML document).

Commands to deal with all the above transformations are accessible in the **XSL/XQuery** menu. In addition, this menu also contains commands to work with the Altova XSLT/XQuery Debugger.

For technical information about the built-in XSLT and XQuery engines, see Altova XQuery Engine, Altova XSLT 1.0 Engine and Altova XSLT 2.0 Engine.

8.6.1 XSL Transformation

F10

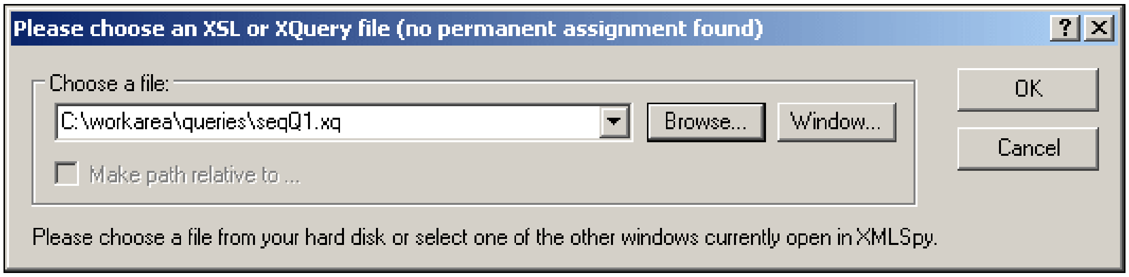
The **XSL Transformation** command transforms an XML document using an assigned XSLT stylesheet. The transformation can be carried out using the appropriate built-in Altova XSLT Engine (Altova XSLT 1.0 Engine for XSLT 1.0 stylesheets; Altova XSLT 2.0 Engine for XSLT 2.0 stylesheets), the Microsoft-supplied MSXML module, or an external XSLT processor. The processor that is used in conjunction with this command is specified in the XSL tab of the Options dialog (**Tools | Options**).

If your XML document contains a reference to an XSLT stylesheet, then this stylesheet is used for the transformation. (An XSLT stylesheet can be assigned to an XML document using the Assign XSL command.) If an XSLT stylesheet has not been assigned to an XML file, you are prompted for te XSLT stylesheet to use.

Also see: Altova XSLT 1.0 Engine and Altova XSLT 2.0 Engine .

8.6.2 XQuery Execution

The **XQuery Execution** command executes an XQuery document. It can be invoked when an XQuery or XML file is active. When invoked from an XML file, it pops up a dialog asking for an XQuery file to associate with the XML file.



Please choose an XSL or XQuery file (no permanent assignment found)

Choose a file: C:\workarea\queries\seqQ1.xq

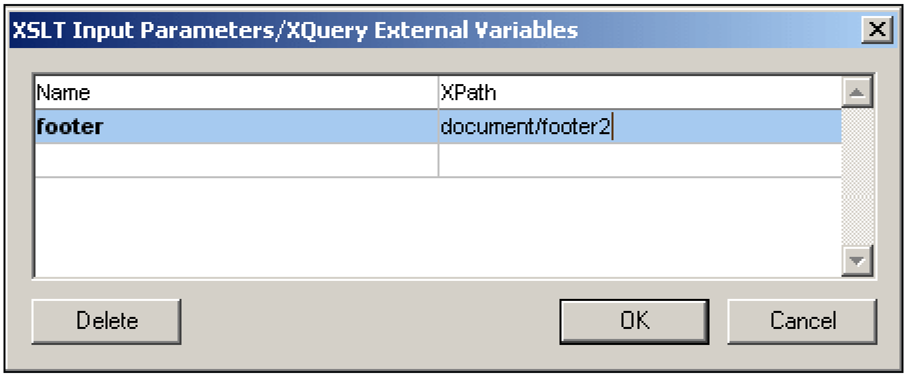
Please choose a file from your hard disk or select one of the other windows currently open in XMLSpy.

8.6.3 XSLT Parameters/XQuery Variables...

The **XSLT Parameters/XQuery Variables** command opens the XSLT Input Parameters/XQuery External Variables dialog (*shown below*). You can enter the name of one or more parameters you wish to pass to the XSLT stylesheet, or one or more external XQuery variables you wish to pass to the XQuery document, and their respective values. The parameter value/s are passed to the parameter/s in the selected XSLT when you process an XML file using the **XSL Transformation** command in the XSL menu. External XQuery variable values are passed to the XQuery document when you select the **XQuery Execution** command.

Using XSLT Parameters

The value you enter for the parameter can be an XPath without quotes or a text string delimited by quotes.



XSLT Input Parameters/XQuery External Variables

Name	XPath
footer	document/footer2

Note: Once a set of parameter-values is entered in the XSLT Input Parameters/XQuery External Variables dialog , they are used for all subsequent transformations until they are explicitly deleted or the application is restarted. Parameters entered in the XSLT Input Parameters/XQuery External Variables dialog are specified at the application-level, and will be passed to the respective XSLT document for every transformation that is carried out via the IDE from that point onward. This means that:

- Parameters are not associated with any particular document, and
- Any parameter entered in the XSLT Input Parameters/XQuery External Variables dialog is erased once the application (XMLSpy 2005) has been closed down.

Using XSLT parameters

In the following example, we select the required document footer from among three possibilities in the XML document (`footer1`, `footer2`, `footer3`).

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
   xsi:noNamespaceSchemaLocation="C:\workarea\footers\footers.xsd">
   <footer1>Footer 1</footer1>
   <footer2>Footer 2</footer2>
   <footer3>Footer 3</footer3>
   <title>Document Title</title>
   <para>Paragraph text.</para>
   <para>Paragraph text.</para>
</document>
```

The XSLT file contains a local parameter called `footer` in the template for the root element. This parameter has a default value of `footer1`. The parameter value is instantiated subsequently in the template with a `$footer` value in the definition of the footer block.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:fo="http://www.w3.org/1999/XSL/Format">
   ...
   <xsl:template match="/">
      <xsl:param name="footer" select="document/footer1" />
      <fo:root>
         <xsl:copy-of select="$fo:layout-master-set" />
         <fo:page-sequence master-reference="default-page"
            initial-page-number="1" format="1">
            <fo:static-content flow-name="xsl-region-after"
               display-align="after">
               ...
               <fo:inline color="#800000" font-size="10pt" font-weight="bold">
                  <xsl:value-of select="$footer"/>
               </fo:inline>
               ...
            </fo:static-content>
         </fo:page-sequence>
      </fo:root>
   </xsl:template>
</xsl:stylesheet>
```

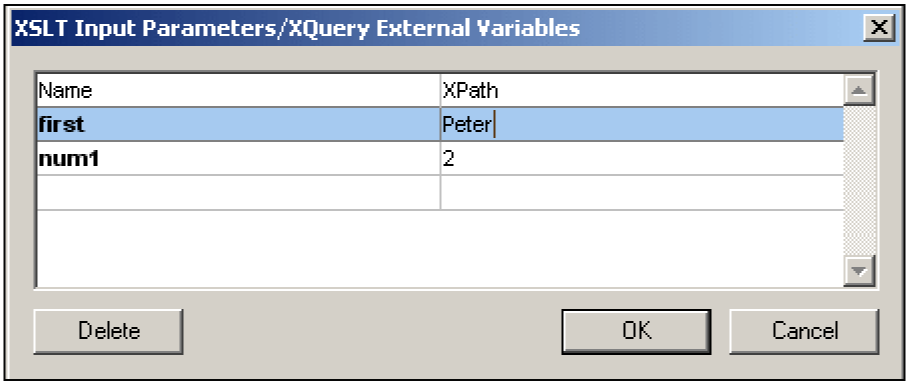
In the XSLT Input Parameters dialog, a new value for the `footer` parameter can be entered, such as the XPath: `document/footer2` (*see screenshot above*) or a text string. During transformation, this value is passed to the `footer` parameter in the template for the root element and is the value used when the footer block is instantiated.

Note:

- If you use the **XSL:FO Transformation** command (**XSL/XQuery | XSL:FO Transformation**), parameters entered in the XSLT Input Parameters/XQuery External Variables dialog are **not** passed to the stylesheet. In order for these parameters to be used in PDF output, first transform from XML to FO using the XSLT Transformation command (**XSL/XQuery | XSL Transformation**), and then transform the FO to PDF using the **XSL:FO Transformation** command (**XSL/XQuery | XSL:FO Transformation**).
- If you use an XSLT processor other than the built-in Altova XSLT Engines, parameters you enter using the XSLT Input Parameters command will not be passed to the external processor.

Using external XQuery variables

The value you enter for an external XQuery variable must be a literal value without quotes. The datatype of the external variable is specified in the variable declaration in the XQuery document.



Name	XPath
first	Peter
num1	2

Note: Once a set of external XQuery variables are entered in the XSLT Input Parameters/XQuery External Variables dialog, they are used for all subsequent transformations until they are explicitly deleted or the application is restarted. Variables entered in the XSLT Input Parameters/XQuery External Variables dialog are specified at the application-level, and will be passed to the respective XQuery document for every execution that is carried out via the IDE from that point onward. This means that:

- Variables are not associated with any particular document, and
- Any variable entered in the XSLT Input Parameters/XQuery External Variables dialog is erased once the application (XMLSpy 2005) has been closed down.

Usage example for external XQuery variables

In the following example, a variable `$first` is declared in the XQuery document with a and is then used in the return clause of the FLWOR expression:

```
xquery version "1.0";
declare variable $first as xs:string external;
let $last := "Jones"
return concat($first, " ", $last )
```

This XQuery returns `Peter Jones`, if the value of the external variable (entered in the XSLT Input Parameters/XQuery External Variables dialog) is `Peter`. Note the following:

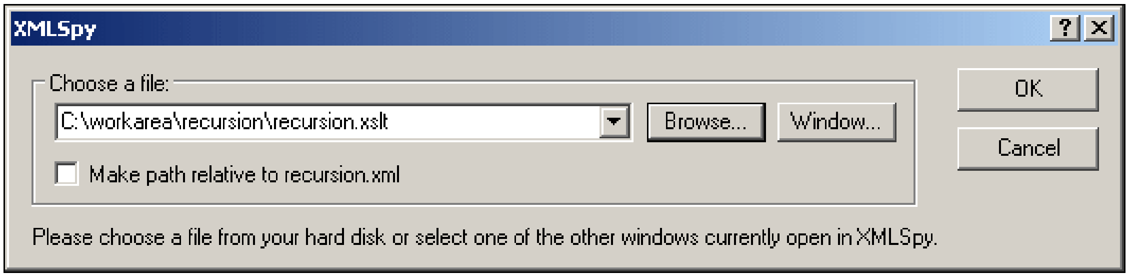
- The `external` keyword in the variable declaration in the XQuery document indicates that this variable is an external variable.
- Defining the static type of the variable is optional. If a datatype for the variable is not specified in the variable declaration, then the variable value is assigned the type `xdt:untypedAtomic`.
- If an external variable is declared in the XQuery document, but no external variable of that name is passed to the XQuery document, then an error is reported.
- If an external variable is declared and is entered in the XSLT Input Parameters/XQuery External Variables dialog, then it is considered to be in scope for the XQuery document being executed. If a new variable with that name is declared within the XQuery document, the new variable temporarily overrides the in-scope external variable. For example, the XQuery document below returns `Paul Jones` even though the in-scope external variable `$first` has a value of `Peter`.

  ```
  xquery version "1.0";
  declare variable $first as xs:string external;
  let $first := "Paul"
  let $last := "Jones"
  return concat($first, " ", $last )
  ```

Note: It is not an error if an external XQuery variable (or XSLT parameter) is defined in the XSLT Input Parameters/XQuery External Variables dialog but is not used in the XQuery document. Neither is it an error if an XSLT parameter (or external XQuery variable) is defined in the XSLT Input Parameters/XQuery External Variables dialog but is not used in an XSLT transformation.

8.6.4 Assign XSL...

The **Assign XSL...** command assigns an XSLT stylesheet to an XML document. Clicking the command opens a dialog to let you specify the XSLT file you want to assign.



Choose a file: C:\workarea\recursion\recursion.xslt

☐ Make path relative to recursion.xml

Please choose a file from your hard disk or select one of the other windows currently open in XMLSpy.

An `xml-stylesheet` processing instruction is inserted in the XML document:

```
<?xml-stylesheet type="text/xsl"
   href="C:\workarea\recursion\recursion.xslt"?>
```

Note that you can make the path of the assigned file relative by clicking the Make Path Relative To... check box.

8.6.5 Assign sample XML file

The **Assign Sample XML File** command assigns an XML file to an XSLT document. The command inserts a processing instruction naming an XML file to be processed with this XSLT file when the XSL Transformation is executed on the XSLT file:

```
<?altova_samplexml C:\workarea\html2xml\article.xml?>
```

Note that you can make the path of the assigned file relative by clicking the Make Path Relative To... check box.

8.7 Authentic Menu

Authentic View enables you to edit XML documents **based on StyleVision Power Stylesheets (.sps files) created in Altova's StyleVision product!** These stylesheets contain information that enables an XML file to be displayed graphically in Authentic View. In addition to containing display information, StyleVision Power Stylesheets also allow you to write data to the XML file. This data is dynamically processed using all the capability available to XSLT stylesheets and instantly produces the output in Authentic View.

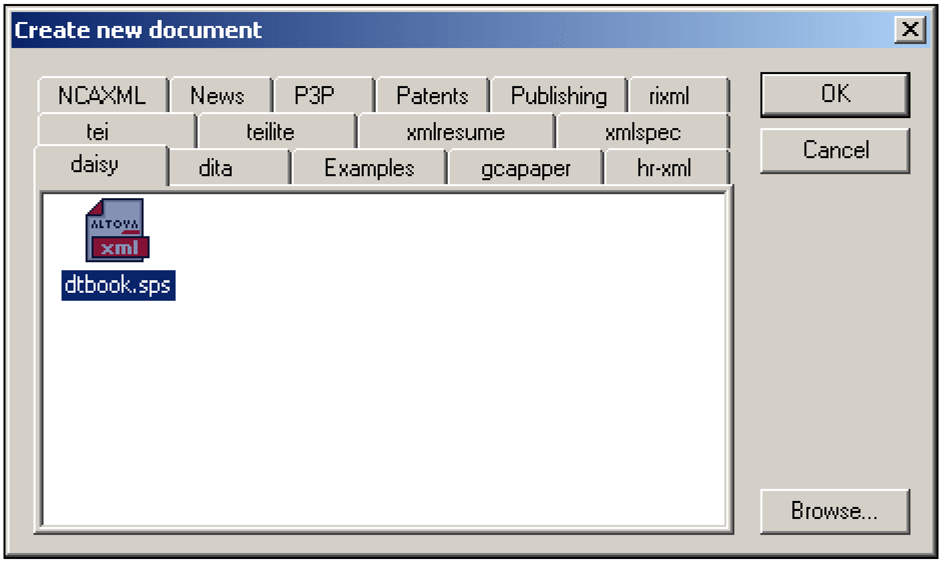
Additionally, StyleVision Power Stylesheets can be created to display an editable XML view of a database. The StyleVision Power Stylesheet contains information for connecting to the database, displaying the data from the database in Authentic View, and writing back to the database.

The Authentic menu contains commands relevant to editing XML documents in Authentic View. For a tutorial on Authentic View, see the Tutorials section.

8.7.1 New Document...

The **New Document...** command enables you to open a new XML document template in Authentic View. The XML document template is based on a StyleVision Power Stylesheet (`.sps` file), and is opened by selecting the StyleVision Power Stylesheet.

Clicking the **New Document...** command pops up the Create New Document dialog.



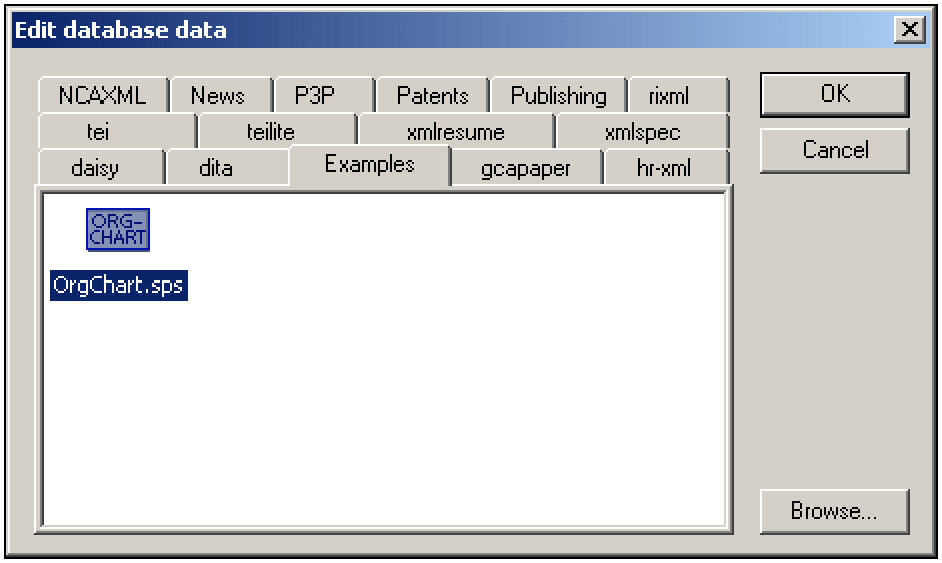
Browse for the required SPS file, and select it. This opens an XML document template in Authentic View.

Note: StyleVision Power Stylesheets are created using Altova StyleVision 2005. The StyleVision Power Stylesheet has a Template XML File assigned to it. The data in this XML file provides the starting data of the new document template that is opened in Authentic View.

8.7.2 Edit Database Data...

The **Edit Database Data...** command enables you to open an editable view of a database (DB) in Authentic View. All the information about connecting to the DB and how to display the DB and accept changes to it in Authentic View is contained in a StyleVision Power Stylesheet. It is such a DB-based StyleVision Power Stylesheet that you open with the **Edit Database Data...** command. This sets up a connection to the DB and displays the DB data (through an XML lens) in Authentic View.

Clicking the **Edit Database Data...** command pops up the Edit Database Data dialog.



Browse for the required SPS file, and select it. This connects to the DB and opens an editable view of the DB in Authentic View. The design of the DB view displayed in Authentic View is contained in the StyleVision Power Stylesheet.

Warning: If, with the **Edit Database Data...** command, you attempt to open a StyleVision Power Stylesheet that is not based on a DB or to open a DB-based StyleVision Power Stylesheet that was created in a version of StyleVision prior to the StyleVision 2005 release, you will receive an error.

Note: StyleVision Power Stylesheets are created using Altova StyleVision 2005.

8.7.3 Assign a StyleVision Stylesheet...

This command assigns a StyleVision Power Stylesheet (SPS) to an **XML document** to enable the viewing and editing of that XML document in Authentic View. The StyleVision Power Stylesheet that is to be assigned to the XML file must be based on the same schema as that on which the XML file is based.

To assign a StyleVision Power Stylesheet to an XML file, make the XML file the active file and select the **Assign a StyleVision Stylesheet...** command. The command opens a dialog box in which you specify the StyleVision Power Stylesheet file you wish to assign to the XML. Clicking OK on your selection inserts the required SPS statement into your XML document. Note that you can make the path to the assigned file relative by clicking the "Make path relative to ..." check box.

```
<?xml version="1.0" encoding="UTF-8"?>
<?altova_sps HTML-Orgchart.sps?>
```

In the example above, the StyleVision Power Stylesheet is called `HTML_Orgchart.sps`, and it is located in the same directory as the XML file.

Note: Previous versions of Altova products used a processing instruction with a target or name of `xmlspysps`, so a processing instruction would look something like `<?xmlspysps HTML-Orgchart.sps?>`. These older processing instructions are still valid with Authentic View in current versions of Altova products.

8.7.4 Edit StyleVision Stylesheet

This command starts StyleVision and allows you to edit the StyleVision Power Stylesheet immediately in StyleVision.

8.7.5 Define XML Entities

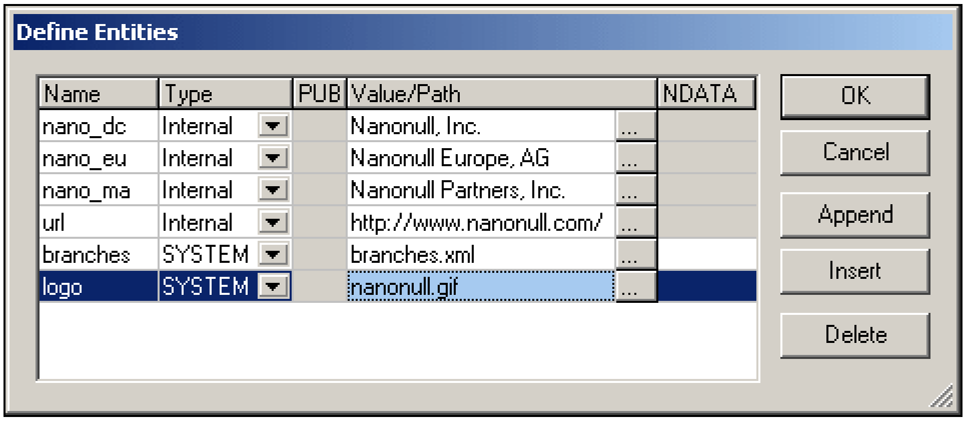
You can define entities for use in Authentic View, whether your document is based on a DTD or an XML Schema. Once defined, these entities are displayed in the Entities Entry Helper and in the **Insert Entity** submenu of the context menu. When you double-click on an entity in the Entities Entry Helper, that entity is inserted at the cursor insertion point.

An entity is useful if you will be using a text string, XML fragment, or some other external resource in multiple locations in your document. You define the entity, which is basically a short name that stands in for the required data, in the Define Entities dialog. After defining an entity you can use it at multiple locations in your document. This helps you save time and greatly enhances maintenance.

There are two broad types of entities you can use in your document: a **parsed entity**, which is XML data (either a text string or a fragment of an XML document), or an **unparsed entity**, which is non-XML data such as a binary file (usually a graphic, sound, or multimedia object). Each entity has a name and a value. In the case of parsed entities the entity is a placeholder for the XML data. The value of the entity is either the XML data itself or a URI that points to a `.xml` file that contains the XML data. In the case of unparsed entities, the value of the entity is a URI that points to the non-XML data file.

To define an entity, do the following:

1. Click **Authentic | Define Entities...**. This opens the Define Entities dialog.



Define Entities

Name	Type	PUB	Value/Path	NDATA
nano_dc	Internal		Nanonull, Inc.	
nano_eu	Internal		Nanonull Europe, AG	
nano_ma	Internal		Nanonull Partners, Inc.	
url	Internal		http://www.nanonull.com/	
branches	SYSTEM		branches.xml	
logo	SYSTEM		nanonull.gif	

2. Enter the name of your entity in the Name field. This is the name that will appear in the Entities Entry Helper.
3. Enter the type of entity from the drop-down list in the Type field. Three types are possible. An **Internal** entity is one for which the text to be used is stored in the XML document itself. Selecting **PUBLIC** or **SYSTEM** specifies that the resource is located

outside the XML file, and will be located with the use of a public identifier or a system identifier, respectively. A system identifier is a URI that gives the location of the resource. A public identifier is a location-independent identifier, which enables some processors to identify the resource. If you specify both a public and system identifier, the public identifier resolves to the system identifier, and the system identifier is used.

4. If you have selected PUBLIC as the Type, enter the public identifier of your resource in the PUBLIC field. If you have selected Internal or SYSTEM as your Type, the PUBLIC field is disabled.
5. In the Value/Path field, you can enter any one of the following:

   - If the entity type is Internal, enter the text string you want as the value of your entity. Do not enter quotes to delimit the entry. Any quotes that you enter will be treated as part of the text string. Note that entities are a good mechanism for including Unicode characters in your document; do this by entering the Unicode number as the value of an internal entity.
   - If the entity type is SYSTEM, enter the URI of the resource or select a resource on your local network by using the Browse button. If the resource contains parsed data, it must be an XML file (i.e. it must have a `.xml` extension). Alternatively, the resource can be a binary file, such as a GIF file.
   - If the entity type is PUBLIC, you must additionaly enter a system identifier in this field.

6. The NDATA entry tells the processor that this entity is not to be parsed but to be sent to the appropriate processor. The NDATA field should therefore be used with unparsed entities only.

Dialog features
You can append, insert, and delete entities by clicking the appropriate buttons. You can also sort entities on the alphabetical value of any column by clicking the column header; clicking once sorts in ascending order, twice in descending order. You can also resize the dialog box and the width of columns.

Limitations

- An entity contained within another entity is not resolved, either in the dialog, Authentic View, or XSLT output, and the ampersand character of such an entity is displayed in its escaped form, i.e. `&amp;`.
- External entities are not resolved in Authentic View, except in the case where an entity is an image file and it is entered as the value of an attribute which has been defined in the schema as being of type `ENTITY` or `ENTITIES`. Such entities are resolved when the document is processed with an XSLT generated from the Authentic Stylesheet.

8.7.6 Hide markup

This command hides markup symbols in the Authentic View.

8.7.7 Show Small markup

This command shows small markup symbols in the Authentic View.

8.7.8 Show Large markup

This command shows large markup symbols in the Authentic View.

8.7.9 Show Mixed markup

This command shows mixed markup symbols in the Authentic View.

8.7.10 Append row

This command appends a row to the current table in the Authentic View.

8.7.11 Insert row

This command inserts a row into the current table in the Authentic View.

8.7.12 Duplicate row

This command duplicates the current table row in the Authentic View.

8.7.13 Move row Up

This command moves current row up by one row in the Authentic View.

8.7.14 Move row Down

This command moves the current row down by one row in the Authentic View.

8.7.15 Delete row

This command deletes the currently active row in the Authentic View.

8.8 View Menu

The View menu controls the display of the active Main window and allows you to change the way XMLSpy 2005 displays your XML documents.

This section provides a complete description of commands in the View menu.

8.8.1 Text View

This command **switches** the current view of the document to Text View, which enables you to edit the document in its text form. It supports a number of advaced text editing features, described in detail in Text View section of this document.

Note: You can configure aspects of the Text View using options available in the various tabs of the Options dialog (**Tools | Options**).

8.8.2 Schema/WSDL Design View

This command **switches** the current document to Schema/WSDL View. This view is described in detail in the Schema/WSDL Design view section of this documentation.

8.8.3 Authentic View

This command **switches** the current document into the Authentic View.

Authentic View enables you to edit XML documents **based on Authentic Stylesheet templates created in StyleVision!** The templates in StyleVision are saved as **Authentic Stylesheet**s (***.sps** files), and supply all the necessary information needed by Authentic View.

Templates are **opened** by Selecting the **File | New** command and then clicking the "Select a STYLEVISION stylesheet..." button. Please see the Authentic View documentation for further information.

Note: If, when you try to switch to Authentic View, you receive a message saying that a temporary (temp) file could not be created, contact your system administrator. The system administrator must change the default Security ID for "non-power users" to allow them to create folders and files.

8.8.4 Browser View

This command **switches** the current document into Browser View.

This view uses an XML-enabled browser (preferably Internet Explorer 5) to render the XML document using information from potential CSS or XSL style-sheets.

When switching to browser view, the document is first checked for validity, if you have selected

Validate upon saving in the File tab of the I Options dialog. Use the menu command **Tools | Options** to open this dialog.

For further information on this view, please see the detailed description of the various views in the Main Window section.

8.8.5 Word Wrap

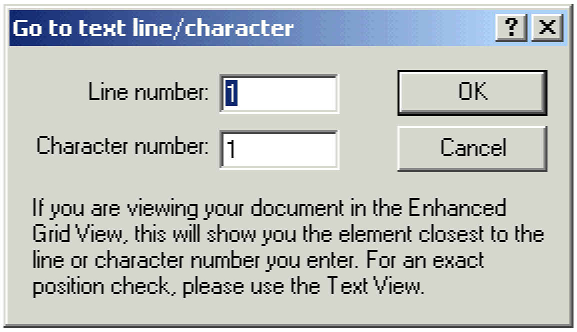
This command enables or disables word wrapping in the **Text view**.

8.8.6 Go to line/char...

Hotkey: **CTRL+G**

This command goes to a **specific line number** and/or character position in an XML document in the Text view.

If you are working with an external XSLT processor (see the XSL page on the Tools | Options dialog for details) you may often get error messages by line number and character position. XMLSpy 2005 lets you quickly navigate to that spot, using this command:



Go to text line/character

Line number: 1

Character number: 1

If you are viewing your document in the Enhanced Grid View, this will show you the element closest to the line or character number you enter. For an exact position check, please use the Text View.

8.8.7 Go to File

This command **opens** a document that is being **referred** to, from within the file you are currently editing.

Select the file name, path name, or URL you are interested in, and choose this command from the View menu.

You can select:

- Some characters from within any item in the Text view.
- An enclosed string. If you text cursor is between quotes, XMLSpy 2005 will automatically use the entire string that is enclosed in the quotes.

8.9 Browser Menu

The **Browser** menu contains commands that are only available in the Browser View.

8.9.1 Back

Backspace

The **Back** command displays the previously viewed page. The Backspace key also achieves the same effect. The **Back** command is useful if you click a link in your XML document and want to return to it.

8.9.2 Forward

The **Forward** command is only available once you have used the **Back** command. It moves you forward through previously viewed pages.

8.9.3 Stop

The **Stop** command instructs the browser to stop loading your document. This is useful if large external files or graphics are being downloaded over a slow Internet connection, and you wish to stop the process.

8.9.4 Refresh

F5

The **Refresh (F5)** command updates the Browser View by reloading the document and related documents, such as CSS and XSL stylesheets, and DTDs.

8.9.5 Fonts

The **Fonts** command allows you to select the default font sie for rendering the text of your XML document. It is similar to the Font Size command in most browsers.

8.9.6 Separate Window

The **Separate Window** command opens the Browser View in a separate window, so that side-by-side viewing is possible. If you have separated the Browser View, press **F5** in editing view to automatically refresh the corresponding Browser View. To dock separate windows back into the interface, click the maximize button (at top right) of the active window.

8.10 Tools Menu

The tools menu allows you to:

- Compare any two files to check for differences
- Compare any two folders to check for differences
- Customize your version: define your own toolbars, keyboard shortcuts, menus, and macros
- Define the global program settings

8.10.1 Spelling...

XMLSpy 2005 now includes a built-in spelling checker.

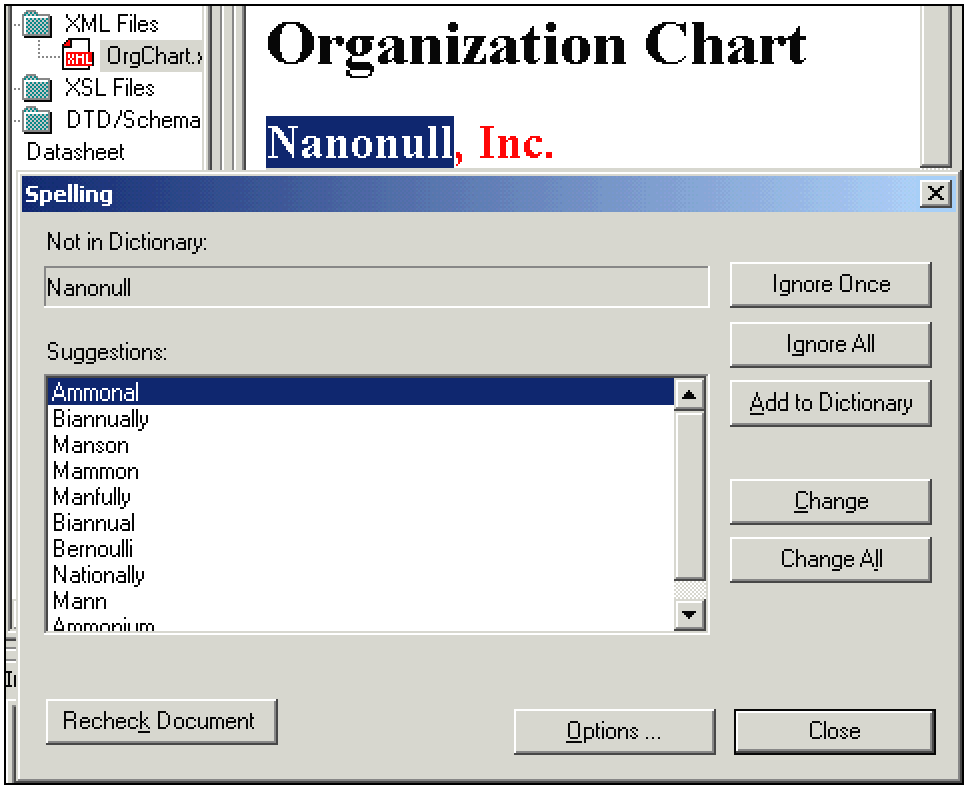
The spelling checker can be used in the following views: Text and Authentic View as well as in StyleVision. Please note that you can check any type of XML file. The view you select influences the spelling checker options that are made available. A schema file (*.xsd) can be checked in the Text view, but not in the Schema/WSDL Design view.

Spell checking in the **Text** view presents you with the most options, you can additionally check the spelling of:

- Element content
- Attribute content
- CDATA content
- Comment text

  You can also define which specific element or attribute content to check or ignore. These selections are defined in the "Spelling context" dialog box.

The **Tools | Spelling...** command (hotkey Shift+F7) opens the Spelling dialog box, and automatically starts checking the currently active XML document (in this case the OrgChart.xml document in the Authentic View).



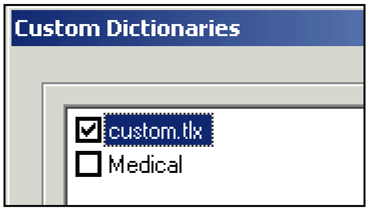
Not in Dictionary
This text box contains the word that cannot be found in any of the existing dictionaries (default and custom dictionaries). You can also edit the text here. The word is also highlighted in the XML document. The command buttons become active at this point and allow you to decide which action to take.

Suggestions
This list box displays a list of words that resemble the unknown word (supplied from all dictionaries). Double clicking a word in this list automatically inserts it in the document and continues the spell checking process.

Ignore once
This command allows you to continue checking the document while ignoring the first occurrence of the unknown word. The same word will be flagged again if it appears in the document.

Ignore all
This command ignores all instances of the unknown word in the whole document.

Add to dictionary
This command adds the unknown word to the currently active **custom dictionary**. This is the dictionary containing the check mark in the Custom Dictionaries dialog box (custom.tlx).



Change
This command replaces the currently highlighted word in the XML document, with the selected word in the Suggestions list box.

Change all
This command replaces all occurrences of the currently highlighted word in the XML document, with the selected word in the Suggestions list box.

Recheck
The Recheck button restarts the check from the beginning of the document.

Close
This command closes the Spelling dialog box.

Options...
This command opens a dialog box depending on the current view.

- If opened from Authentic View, the Options dialog box is opened.
- If opened from the Text view, the Spelling context dialog box is opened.

8.10.2 Spelling options...

This command opens either the **Options** or **Spelling context** dialog box, depending on the view you are using to display your XML document.

Options - **Authentic View** and **StyleVision**
When viewing an XML document in Authentic View, this command opens the Options dialog box. Use this dialog box to define the global spelling checker options.

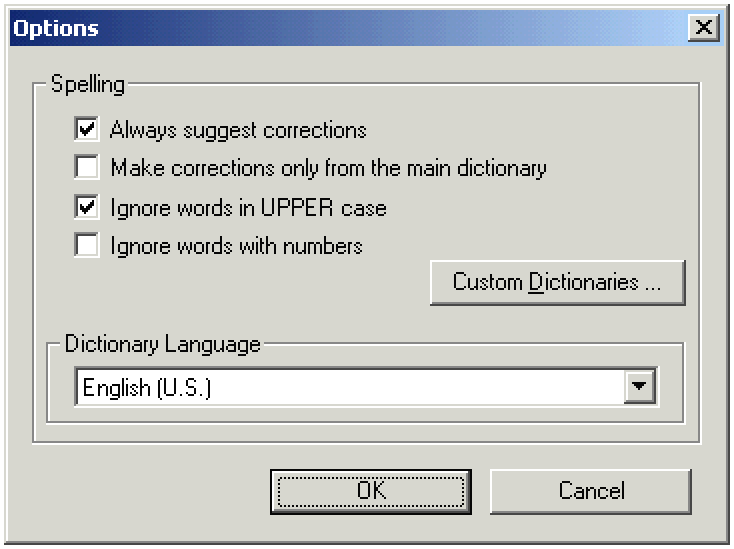
Always suggest corrections:
Activating this option causes suggestions (from all of the dictionaries) to be displayed in the Suggestions list box. Disabling this option causes no suggestions to be shown.

Make corrections only from main dictionary:
Activating this option causes only the default dictionary to be used, none of the custom dictionaries are scanned for suggestions. It also disables the Custom Dictionaries button, preventing any editing of the custom dictionaries.

Ignore words in UPPER case:
Activating this option causes all upper case words to be ignored.

Ignore words with numbers:
Activating this number causes all words containing numbers to be ignored.

Dictionary Language
Use this combo box to select the dictionary language for the spelling checking. The default installation allows you to select English. Other language dictionaries will be made available on the Altova download web site.



☑ Always suggest corrections
☐ Make corrections only from the main dictionary
☑ Ignore words in UPPER case
☐ Ignore words with numbers

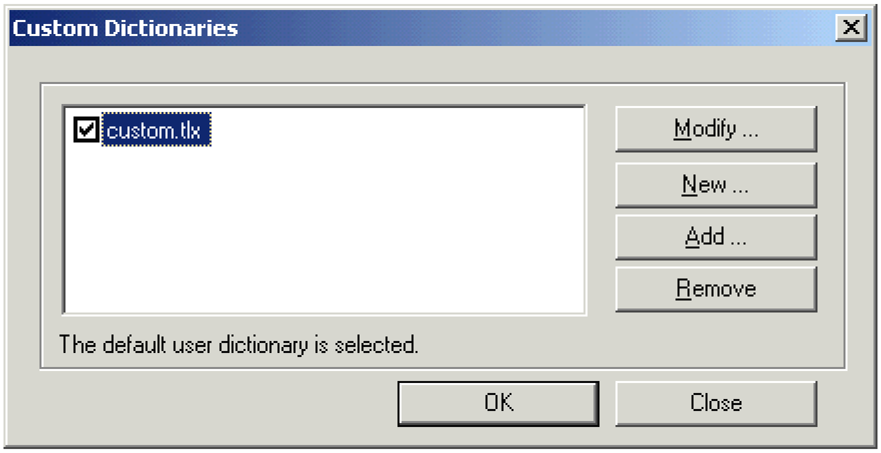
Dictionary Language: English (U.S.)

Custom dictionaries

The Custom Dictionaries... button allows you to:

- modify an existing dictionary (add or delete dictionary entries)
- create a totally new dictionary
- add an existing dictionary
- remove an existing dictionary

When you start the spell checking process, all dictionaries listed in the Custom Dictionaries list box are searched. If you want to limit the search to specific dictionaries, use the Remove command to delete those you do not want searched.

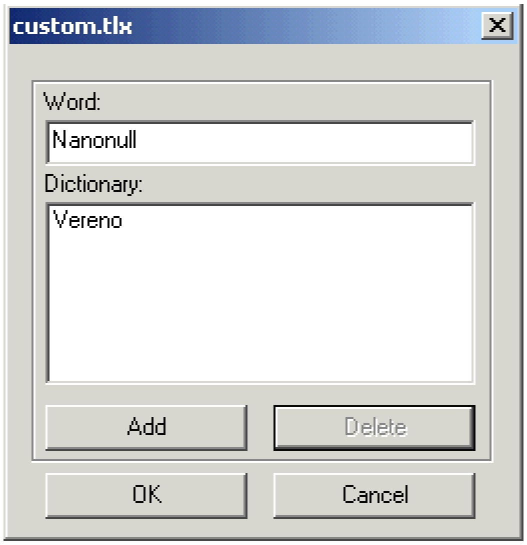


☑ custom.tlx

The default user dictionary is selected.

To modify dictionary entries:

1. Click the custom dictionary name whose entries you want to change, and click Modify... This opens the dictionary highlighted in the list box (custom.tlx in this case). A prompt appears if none of the dictionaries has been selected.



Word: Nanonull

Dictionary: Vereno

2. Click in the Word: field and enter the new dictionary entry (this automatically activates the Add button).
3. Click Add to add it to the dictionary.

To delete an entry from the dictionary:

1. Click the word in the Dictionary list box to highlight it, and click Delete.
2. Click OK to confirm the changes made in this dialog box.

To add a new dictionary:

1. Click New, and enter the name of the new **custom dictionary** in the File name... field.
2. Click Save to save the dictionary.
3. You can now add entries to the dictionary using the Add button, or the Add to Dictionary button while performing a spell check.

To add an existing dictionary:

Use this option to add previously removed (or third party dictionaries).

1. Click Add and select a dictionary from the list box. Dictionaries have a *.**tlx** extension.
2. Click Open to add the dictionary to the Custom dictionary list.

Please note:

It is not mandatory for a dictionary to have a *.tlx extension. It can have any, or no extension at all, and still be added to the dictionary list box.

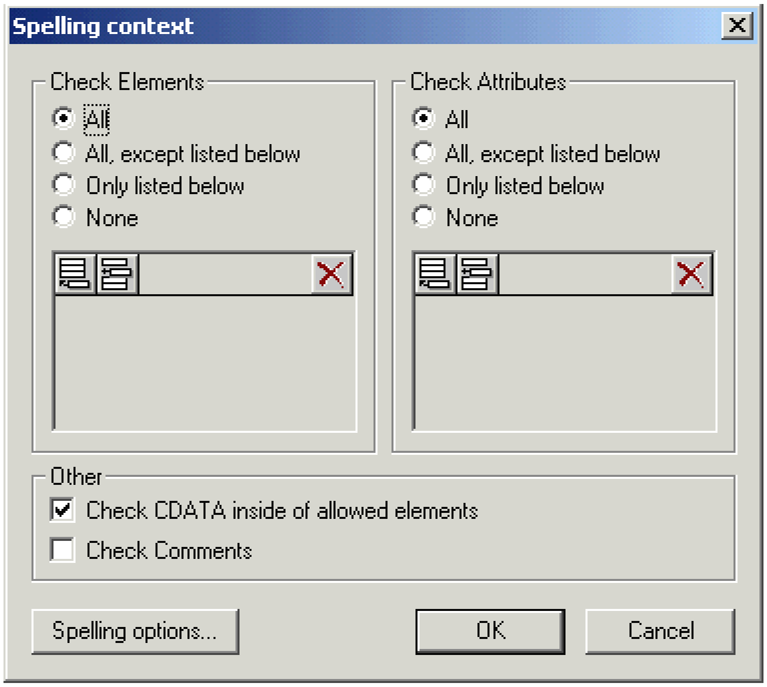
To remove a dictionary:

- Click the dictionary name in the list and click Remove. This removes the dictionary from the list, it does not physically delete it from your hard disk.

Spelling context - **Text** view

When viewing an XML document in the Text view, this command opens the "Spelling context" dialog box. Starting the spelling checking in the **Text** view presents you with the most options, you can define the specific content which is to be checked:

- Element content
- Attribute content
- CDATA content
- Comment text



Spelling context

Check Elements
● All
○ All, except listed below
○ Only listed below
○ None

Check Attributes
● All
○ All, except listed below
○ Only listed below
○ None

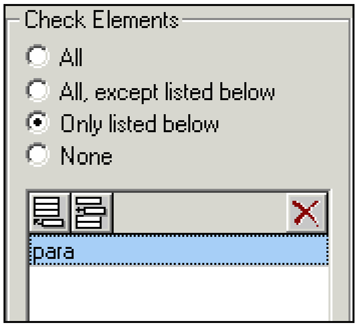
Other
☑ Check CDATA inside of allowed elements
□ Check Comments

The Spelling options.. button opens the Options dialog box, which allows you to define the global spelling checker options.

To define specific element or attribute "content" you want to check:

1. Click the "All, except listed below", or "Only listed below" radio button.

2. Click the Append element/attribute icon , and enter the tag name containing the text you want checked (e.g. para). Hit return to confirm.



Check Elements
○ All
○ All, except listed below
● Only listed below
○ None

para

3. Click the OK button to confirm these settings, and click it again in the Spelling dialog box to start the spelling checker with these new settings.
The only **element content** to be checked will be the text between the **para** tags, attribute content will also be checked.

To delete a tag name:

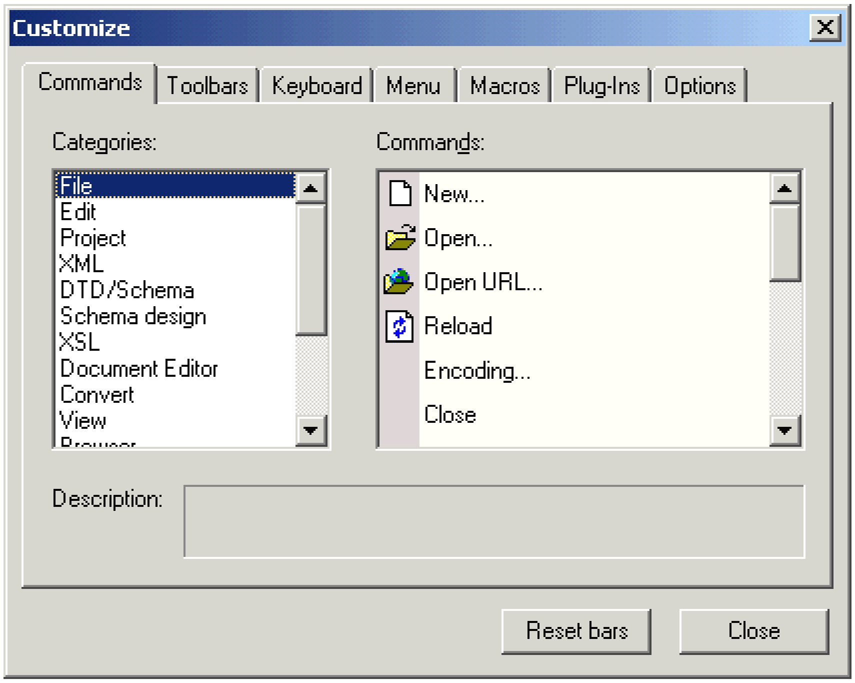
- Select the tag name, and hit the Delete or Backspace key.

8.10.3 Customize...

The customize command lets you customize XMLSpy 2005 to suit your personal needs.

Commands

The Commands tab allows you customize your menus or toolbars.



To add a command to a toolbar or menu:

1. Open this dialog box using **Tools | Customize.**
2. Select the command category in the Categories list box. The commands available appear in the Commands list box.
3. Click on a command in the commands list box and drag "it" to an to an existing menu or toolbar.
4. An **I**-beam appears when you place the cursor over a valid position to drop the command.
5. Release the mouse button at the position you want to insert the command.

- A small button appears at the tip of mouse pointer when you drag a command. The check mark below the pointer means that the command cannot be dropped at the current cursor position.
- The check mark disappears whenever you can drop the command (over a tool bar or menu).
- Placing the cursor over a menu when dragging, opens it, allowing you to insert the command anywhere in the menu.
- Commands can be placed in menus or tool bars. If you created you own toolbar you can populate it with your own commands/icons.

Please note:

You can also edit the commands in the context menus (right click anywhere opens the context menu), using the same method. Click the Menu tab and then select the specific context menu available in the Context Menus combo box.

To delete a command or menu:

1. Open this dialog box using **Tools | Customize.**
2. Click on the menu entry or icon you want to delete, and drag with the mouse.

3. Release the mouse button whenever the check mark icon appears below the mouse pointer.
   The command, or menu item is deleted from the menu or tool bar.

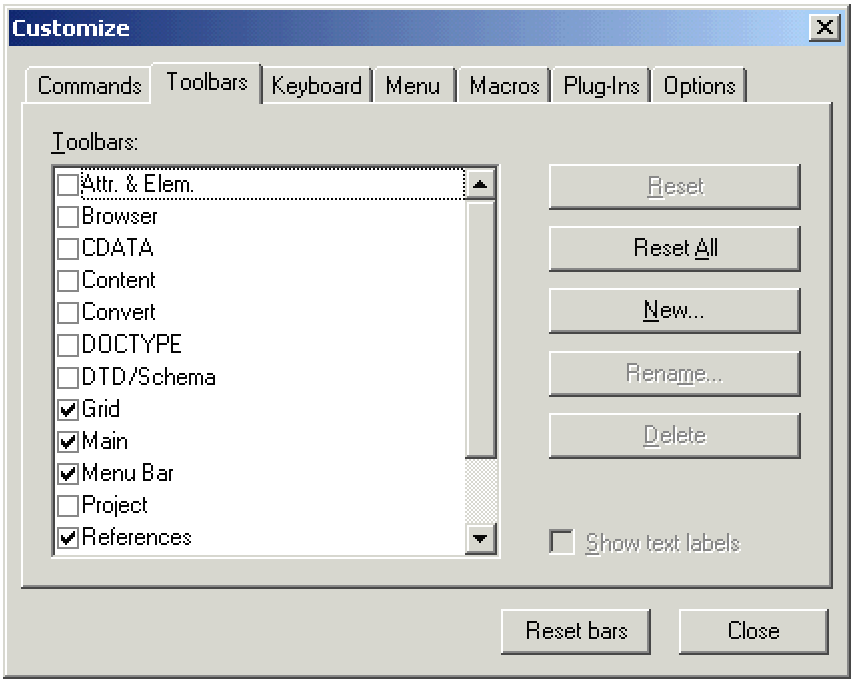
Toolbars

The Toolbars tab allows you to activate or deactivate specific toolbars, as well as create your own specialized ones.

XMLSpy 2005 toolbars contain symbols for the most frequently used menu commands.
For each symbol you get a brief "tool tip" explanation when the mouse cursor is directly over the item and the status bar shows a more detailed description of the command.

You can drag the toolbars from their standard position to any location on the screen, where they appear as a floating window. Alternatively you can also dock them to the left or right edge of the main window.

- Toolbar settings defined in the Schema/WSDL design and Text view are valid in those views. The Browser view toolbars are independent of all the other views.



Toolbars:

☐ Attr. & Elem.
☐ Browser
☐ CDATA
☐ Content
☐ Convert
☐ DOCTYPE
☐ DTD/Schema
☑ Grid
☑ Main
☑ Menu Bar
☐ Project
☑ References

☐ Show text labels

To activate or deactivate a toolbar:

1. Click the check box to activate (or deactivate) the specific toolbar.

To create a new toolbar:

1. Click the **New...** button, and give the toolbar a name in the Toolbar name dialog box.
2. Add commands to the toolbar using the Commands tab of the Customize dialog box.

To reset the Menu Bar

- Click the Menu Bar entry and
- Click the **Reset** button, to reset the menu commands to the state they were in when XMLSpy 2005 was installed.

To reset all toolbar and menu commands

- Click the **Reset All** button, to reset all the toolbar commands to the state they were when the program was installed. A prompt appears stating that all toolbars and menus will be reset.
- Click Yes to confirm the reset.

To change a toolbar name:

- Click the **Rename**... button to edit the name of the toolbar.

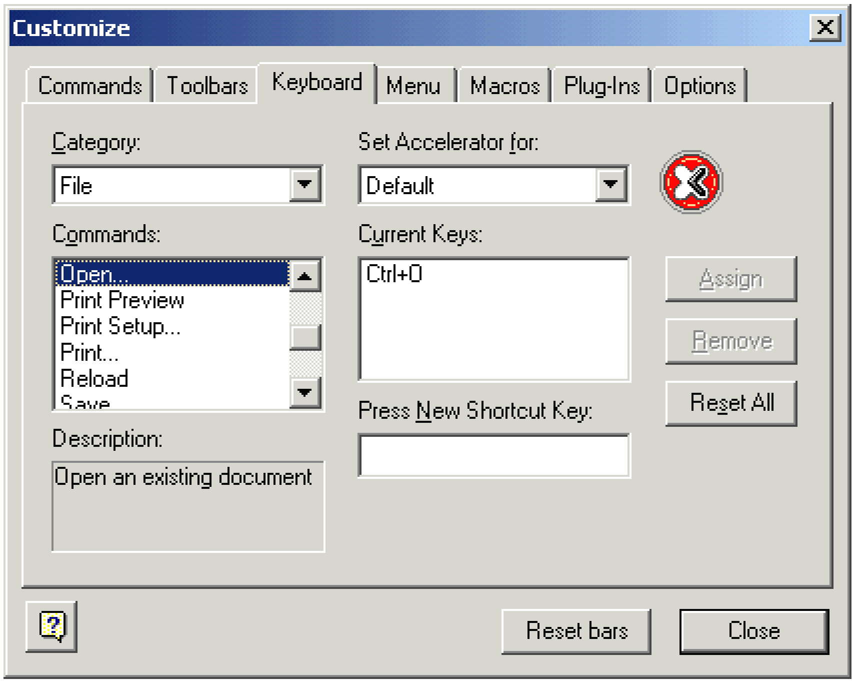
To delete a toolbar:

- Click the **Delete** button to delete the currently highlighted toolbar in the Toolbars list box.
- A prompt appears, asking if you really want to delete the toolbar. Click Yes to confirm the deletion.
- or, click right on the menu or icon bar, and select the Delete entry in the popup menu.

Show text labels:
This option places explanatory text below toolbar icons when activated.

Keyboard

The Keyboard tab allows you to define (or change) keyboard shortcuts for any XMLSpy 2005 command.



Category: File

Set Accelerator for: Default

Commands: Open..., Print Preview, Print Setup..., Print..., Reload, Save

Current Keys: Ctrl+O

Press New Shortcut Key:

Description: Open an existing document

To assign a new Shortcut to a command:

1. Select the commands category using the **Category** combo box.
2. Select the **command** you want to assign a new shortcut to, in the Commands list box
3. Click in the "**Press New Shortcut Key**:" text box, and press the shortcut keys that are to activate the command.
   The shortcuts appear immediately in the text box. If the shortcut was assigned previously, then that function is displayed below the text box.
4. Click the **Assign** button to permanently assign the shortcut.
   The shortcut now appears in the Current Keys list box.
   (To **clear** this text box, press any of the control keys, CTRL, ALT or SHIFT).

To de-assign (or delete a shortcut):

1. Click the shortcut you want to delete in the Current Keys list box, and
2. Click the **Remove** button (which has now become active).
3. Click the Close button to confirm all the changes made in the Customize dialog box.

Set accelerator for:

Currently no function.

Currently assigned keyboard shortcuts:

Hotkeys by key	
F1	Help Menu
F3	Find Next
F5	Refresh
F7	Check well-formedness
F8	Validate
F9	Insert/Remove breakpoint
CTRL+F9	Enable/Disable breakpoint
F10	XSL Transformation
CTRL+F10	XSL:FO Transformation
F11	Step into
CTRL+F11	Step Over
Shift + F11	Step Out
Alt+F11	Start Debugger/Go
Num +	Expand
Num -	Collapse
Num *	Expand fully
CTRL+Num-	Collapse unselected
CTRL + G	Goto line/char
CTRL+TAB	and
CTRL+F6	Cycle through open windows
Arrow keys	
(up / down)	Move selection bar
Esc.	Abandon edits/close dialog box
Return/Space bar	confirms a selection
Alt + F4	Closes XMLSpy 2005
CTRL + F4	Closes active window
Alt + F, 1	Open last file

CTRL + Double click an element (Schema view)	Display element definition
CTRL + N	File New
CTRL + O	File Open
CTRL + S	File Save
CTRL + P	File Print
CTRL + A	Select All
Shift + Del	Cut (or CTRL + X)
CTRL + C	Copy
CTRL + V	Paste
CTRL + Z	Undo
CTRL + Y	Redo
Del	Delete (Delete item in Schema/)
CTRL + F	Find
F3	Find Next
CTRL + H	Replace
CTRL + I	Append Attribute
CTRL + E	Append Element
CTRL + T	Append Text
CTRL + D	Append CDATA
CTRL + M	Append Comment
CTRL + SHIFT + I	Insert Attribute
CTRL + SHIFT + E	Insert Element
CTRL + SHIFT + T	Insert Text content
CTRL + SHIFT + D	Insert CDATA
CTRL + SHIFT + M	Insert Comment
CTRL + ALT + I	Add Child Attribute
CTRL + ALT + E	Add Child Element
CTRL + ALT + T	Add Child Text
CTRL + ALT + D	Add Child CDATA
CTRL + ALT + M	Add Child Comment

Currently assigned keyboard shortcuts:

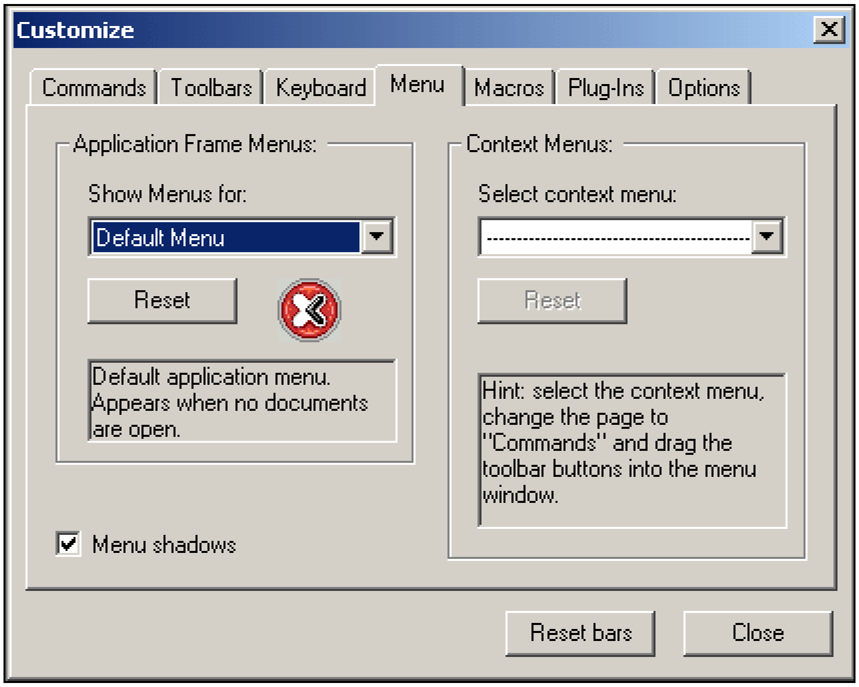
Hotkeys by function

Abandon edits	Esc.
Add Child Attribute	CTRL + ALT + I
Add Child CDATA	CTRL + ALT + D
Add Child Comment	CTRL + ALT + M
Add Child Element	CTRL + ALT + E
Add Child Text	CTRL + ALT + T
Append Attribute	CTRL + I
Append CDATA	CTRL + D
Append Comment	CTRL + M
Append Element	CTRL + E
Append Text	CTRL + T
Check well-formedness	F7
Closes active window	CTRL + F4
Close XMLSpy 2005	Alt + F4
Collapse	Num -
Collapse unselected	CTRL + Num-
Confirms a selection	Return / Space bar
Copy	CTRL + C
Cut	SHIFT + Del (or CTRL + X)
Cycle through windows	CTRL + TAB and CTRL + F6
Delete item	Del
Enable/Disable breakpoint	CTRL + F9
Expand	Num +
Expand fully	Num *
File New	CTRL + N
File Open	CTRL + O
File Print	CTRL + P
File Save	CTRL + S

Find	CTRL + F
Find Next	F3
Goto line/char	CTRL + G
Help Menu	F1
Insert Attribute	CTRL + SHIFT + I
Insert CDATA	CTRL + SHIFT + D
Insert Comment	CTRL + SHIFT + M
Insert Element	CTRL + SHIFT + E
Insert/Remove breakpoint	F9
Insert Text content	CTRL + SHIFT + T
Move selection bar	Arrow keys (up / down)
Open last file	Alt + F, 1
Paste	CTRL + V
Redo	CTRL + Y
Refresh	F5
Replace	CTRL + H
Select All	CTRL + A
Start Debugger/Go	Alt + F11
Step Into	F11
Step Out	Shift + F11
Step Over	CTRL + F11
To view an element definition	CTRL + Double click on an element.
Undo	CTRL + Z
Validate	F8
XSL Transformation	F10
XSL:FO Transformation	CTRL + F10

Menu

The Menu tab allows you to customize the main menu bars as well as the (popup - right click) context menus.



Application Frame Menus:

Show Menus for: Default Menu

Default application menu. Appears when no documents are open.

Context Menus:

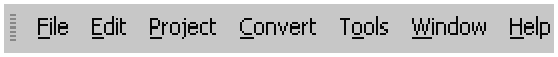
Select context menu: --

Hint: select the context menu, change the page to "Commands" and drag the toolbar buttons into the menu window.

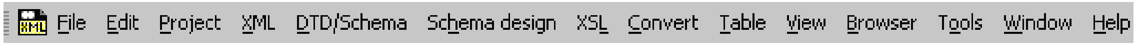
☑ Menu shadows

You can customize both the Default and XMLSpy 2005 menu bars.

The **Default** menu is the one visible when no XML documents of any type are open in XMLSpy 2005.



The XMLSpy 2005 menu is the menu bar visible when at least one XML document has been opened.



To customize a menu:

1. Select the menu bar you want to customize from the "Show Menus for:" combo box
2. Click the **Commands** tab, and drag the commands to the menu bar of your choice.

To delete commands from a menu:

1. Click right on the command, or icon representing the command, and
2. Select the **Delete** option from the popup menu,

or,

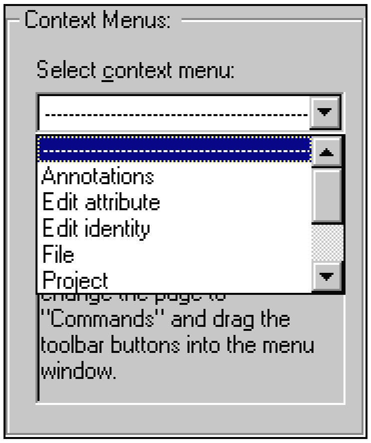
1. Select **Tools | Customize** to open the Customize dialog box, and
2. Drag the command away from the menu, and drop it as soon as the check mark icon appears below the mouse pointer.

To reset either of the menu bars:

1. Select either the Default or XMLSpy 2005 entry in the combo box, and
2. Click the **Reset** button just below the menu name.
   A prompt appears asking if you are sure you want to reset the menu bar.

To customize any of the Context menus (right click menus):

1. Select the context menu from the combo box.
2. Click the **Commands** tab, and drag the commands to context menu that is now open.



change the page to "Commands" and drag the toolbar buttons into the menu window.

To delete commands from a context menu:

1. Click right on the command, or icon representing the command, and
2. Select the **Delete** option from the popup menu

or,

1. Select **Tools | Customize** to open the Customize dialog box, and
2. Drag the command away from the context menu, and drop it as soon as the check mark icon appears below the mouse pointer.

To reset any of the context menus:

1. Select the context menu from the combo box, and
2. Click the **Reset** button just below the context menu name.
   A prompt appears asking if you are sure you want to reset the context menu.

To close an context menu window:

1. Click on the **Close icon** at the top right of the title bar, or
2. Click the Close button of the Customize dialog box.

Menu animations (only prior to Windows 2000)

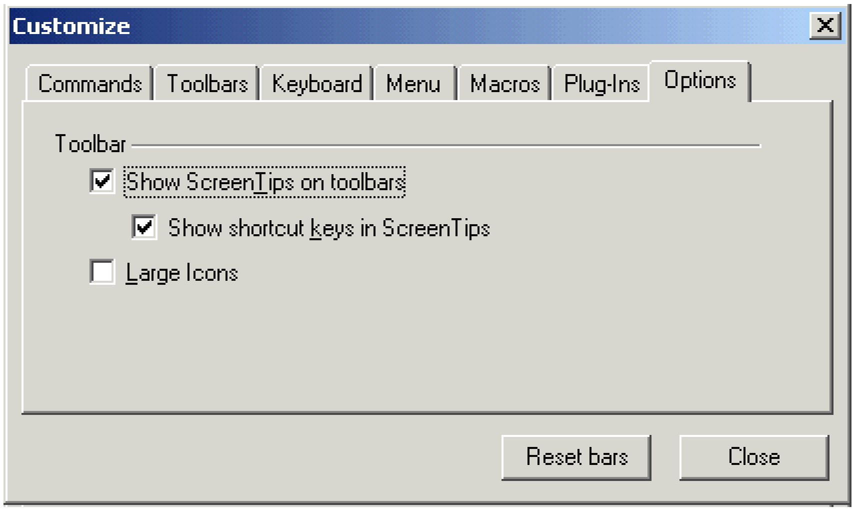
- Select one of the menu animations from the combo box, if you want animated menus. Please note that the combo box is only visible in Windows versions prior to Windows 20000. For Windows 2000 and later, these settings have to be changed in the Effects tab of the Display properties dialog box. Double click the Display icon in the Control Panel to open the dialog box.

Menu shadows

- Click the Menu shadows check box, if you want all your menus to have shadows.

Options

The Options tab allows you to set general environment settings.



Toolbar

☑ Show ScreenTips on toolbars

☑ Show shortcut keys in ScreenTips

☐ Large Icons

Toolbar
When active, the **Show Tooltips on toolbars** check box displays a popup when the mouse pointer is placed over an icon in any of the icon bars. The popup contains a short description of the icon function, as well as the associated keyboard shortcut, if one has been assigned.

The **Show shortcut keys in Tooltips** check box, allows you to decide if you want to have the shortcut displayed in the tooltip.

When active, the **Large icons** check box switches between the standard size icons, and larger versions of the icons.

8.10.4 Options

The **Options** command enables you to define global application settings. These settings are specified in a tabbed dialog box and saved in the registry. They apply to all current and future document windows. The **Apply** button displays the changes in the currently open documents and fixes the current settings. The changes are seen immediately in the background windows.

Each tab of the Options dialog is described in detail in this section.

File

The **File** tab defines the way XMLSpy 2005 opens and saves documents. Related settings are in the Encoding tab.

Automatic reload of changed files
If you are working in a multi-user environment, or if you are working on files that are dynamically generated on a server, you can watch for changes to files are currently open in the interface. Each time XMLSpy 2005 detects a change in an open document, it will prompt you about whether you want to reload the changed file.

Validation
If you are using DTDs or schemas to define the structure of your XML documents, you can automatically check the document for validity whenever it is opened or saved. XMLSpy 2005 can also cache these files in memory to save any unnecessary reloading (e.g. when the

Schema being referred to, is accessed through a URL). If your schema location declaration uses an URL, disable the "cache DTD/Schema files in memory" option, to have changes made to the schema appear immediately, and not use the cached version of the schema., you can open the last-used project automatically.

Save File
When the file is saved, indentation will be saved as specified in this pane, i.e. with tab characters (`#x09`), two spaces, or without any indentation. If a StyleVision Power Stylesheet is associated with an XML file, the 'Authentic: save link to design file' option will cause the link to the StyleVision Power Stylesheet to be saved with the XML file. Note that the indentation you specify here is also used when the document is formatted using the pretty-print feature (**Edit | Pretty-Print XML Text**).

When saving an XML document, XMLSpy 2005 includes a short comment `<!-- Edited with XMLSpy 2005 http://www.altova.com -->` near the top of the file. This option can only be deactivated by licensed users.

When saving a content model diagram (using the menu option **Schema design | Generate Documentation**), XMLSpy 2005 includes the XMLSpy 2005 logo. This option can only be deactivated by licensed users.

Line breaks
When you open a file, the character coding for line breaks in it are preserved if **Preserve old** is selected. Alternatively, you can choose to code line breaks in any of three codings: **CR&LF** (for PC), **CR** (for MacOS), or **LF** (for Unix).

No output formatting for
In Text View, child elements can be indented (with tabs or spaces) for easier readability. When indentation is specified, the file is saved with the specified whitespace characters. The indentation of an element can also be made to reflect its position in the document hierarchy (**Edit | Pretty-Print XML Text**). You can, however, override the indentation you have specified (either in this dialog or manually in the file) for individual elements by entering the element name in the **No output formatting for** field. All elements entered in this field will be saved (and displayed) without whitespace between consecutive descendant elements.

File types

The **File types** tab allows you to customize the behavior of XMLSpy 2005 on a per-file-type basis.

Choose a file type from the File Types list box to customize the functions for that particular file type:

Windows Explorer settings
You can define the file type description and MIME-compliant content type used by Windows Explorer and whether XMLSpy 2005 is to be the default editor for documents of this file type.

Conformance
XMLSpy 2005 provides specific editing and other features for various file types. The features for a file type are set by specifying the conformance in this option. XMLSpy 2005 lets you set file type to conform with XML, XQuery, and other (text) grammars. Furthermore, XML conformance is differentiated between XML, DTD, and XML Entity file types. A large number of file types are defined with a default conformance that is appropriate for the file type. We recommend that you do not modify these settings unless you are adding a new file type or deliberately wish to set a

file type to another kind of conformance.

Default view
This group lets you define the default view to be used for each file type.

Text View
This text box lets you set syntax-coloring for particular file types.

Disable automatic validation
This option enables you to disable automatic validation per file type. Automatic validation typically takes place when a file is opened or saved, or when a view is changed.

Save empty elements in short <E/> format
Some applications that use XML documents or output generated from XML documents may have problems understanding the short `<Element/>` form for empty elements defined in the XML 1.0 Specification. You can instruct XMLSpy 2005 to save elements in the longer (but also valid) `<Element></Element>` form.

Add new file extension
Adds a new file type to the File types list. You must then define the settings for this new file type using the other options in this tab.

Delete selected file extension
Deletes the currently selected file type and all its associated settings.

Editing

The **Editing** tab enables you to specify editing behaviour in XMLSpy 2005.

Intelligent editing
While editing documents, XMLSpy 2005 provides Intelligent Editing based on these settings. You can also customize various aspects of the behavior of these Entry Helpers here.

View

The **View** tab enables you to customize the XML documents presentation in XMLSpy 2005.

Text View
Turn the Text View word-wrapping support off.

Program logo
You can turn off the splash screen upon program startup to speed up the application.

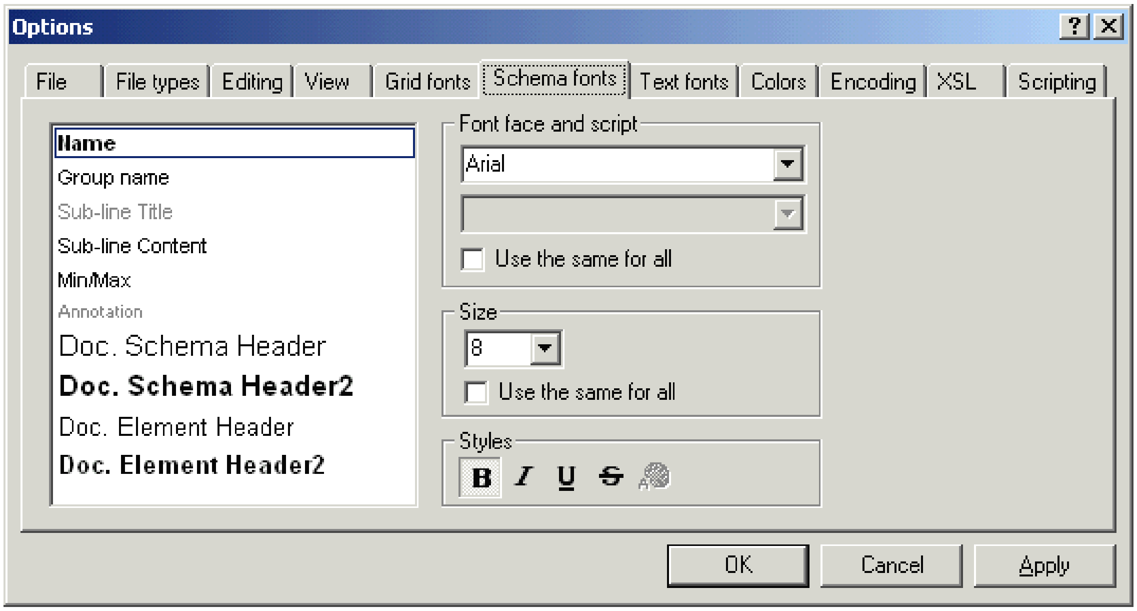
Window title
The window title for each document window can contain either the file name only or the full path name.

Authentic View
XML files based on a **StyleVision Power Stylesheet** are automatically opened in the Authentic View when this option is active.

Browser View
You can choose to see the browser view in a separate window, enabling side-by-side placement of the edit and browser views.

Schema fonts

The **Schema fonts** tab enables you to customize the appearance of text in the Schema/WSDL View.



Font face and script: Arial

□ Use the same for all

Size: 8

□ Use the same for all

Font face and script

You can select the font face and size to be used for displaying the various items in the Schema/WSDL Design view. The same fonts are used when printing and creating schema documentation. Consequently only TrueType fonts should be selected. Components prefixed with "Doc." are used in the schema documentation.

Size

Select the required size. If you want to use the same font size for all items, click on the Use The Same For Al" check box.

Styles

The style and color can be set using the options in this pane. The current settings are immediately reflected in the list in the left pane, so you can preview the way your document will look.

Text fonts

The **Text fonts** tab you to customize the appearance of text in the Text View.

The types listed in the left hand pane are XML node types, ASP/JSP code, and XQuery grammar components. You can choose the common font face, style and size of all text that appears in Text View. Note that the same font, style, and size is used for all text types. Only the text color and background color can be changed for individual text types. This enables the syntax coloring feature.

Encoding

The **Encoding** tab specifies options for file encodings.

Default encoding for new XML files

The default encoding for new files can be pre-determined in the Settings dialog box so that each new document is automatically created with a proper XML-declaration and includes the encoding-specification that you specify here. If a two- or four-byte encoding is selected as the default encoding (i.e. UTF-16, UCS-2, or UCS-4) you can also choose between little-endian and big-endian byte-ordering for the XML files. The Default Encoding for New XML Files setting only works for files which do not have a StyleVision Power Stylesheet associated with it.

The encoding for existing files will, of course, always be retained and can only be changed with the **File | Encoding** command.

Open XML files with unknown encoding as
You can select the encoding with which to open an XML file with no encoding specification or where the encoding cannot be detected. Note that XML files which have no encoding specification are correctly saved with a UTF-8 encoding.

Open non-XML files in
You can select the encoding of non-XML files that you wish to edit.

8.11 ADVANCED Menu

Clicking the **ADVANCED** Menu opens the **Try an Advanced Feature for the Day** dialog. In this dialog, you can select one of the several advanced features which are listed on the left-hand side of the dialog. These advanced features are available in the Enterprise or Professional Editions of XMLSpy 2005 but not in the Home Edition. You can try out each of these features for one calendar day. Only one feature can be tried out at a time, and each feature can be tried out a limited number of times.

View a description of an Advanced Feature
To view a description of an Advanced Feature, click the radio button next to it. The description appears in the main window of the dialog.

Trying out an Advanced Feature
To try out an Advanced Feature, click the **Try Now!** button at the bottom right of the dialog. (You will need to be connected to the Internet so that a request can be sent to the Altova server.) Shortly afterwards, you will be prompted to re-start XMLSpy 2005. When you re-start, the new feature will be available. New menus and/or menu items and their associated dialogs will be available; shortcut icons relevant to the Advanced Feature may appear in the toolbar; and additional or modified views may appear in the Main Window.

If you would like to use one or more Advanced Feature/s, you can purchase a key-code for the appropriate edition (Enterprise or Professional) at the Altova Website.

Installing XMLSpy 2005 Enterprise or Professional Edition

1. Download the self-contained installer from the Altova website.
2. Purchase a key-code for your product from the Altova Online Shop at the Altova website.
3. Start the installer (by double-clicking it). The Home Edition will automatically be uninstalled before the new Enterprise or Professional Edition is installed.

8.12 Window Menu

To organize the individual document windows in an XMLSpy 2005 session, the Window menu contains standard commands common to most Windows applications.

You can cascade the open document windows, tile them, or arrange document icons once you have minimized them. You can also switch the various Entry Helper windows on or off, or switch to an open document window directly from the menu.

8.12.1 Cascade

This command rearranges all open document windows so that they are all cascaded (i.e. staggered) on top of each other.

8.12.2 Tile horizontally

This command rearranges all open document windows as **horizontal tiles**, making them all visible at the same time.

8.12.3 Tile vertically

This command rearranges all open document windows as **vertical tiles**, making them all visible at the same time.

8.12.4 Info Window

This command lets you switch the Info Window on or off.

This is a dockable window. Dragging on its title bar detaches it from its current position and makes it a floating window. Click right on the title bar, to allow docking or hide the window.

8.12.5 Entry Helpers

This command lets you switch all three Entry-Helper Windows on or off.

All three Entry helpers are dockable windows. Dragging on a title bar detaches it from its current position and makes it a floating window. Click right on the title bar, to allow docking or hide the window.

8.12.6 All on/off

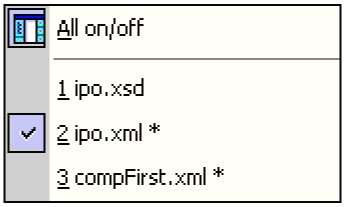
This command lets you switch all dockable windows on, or off:

- the Info Window
- the three Entry-Helper Windows

This is useful if you want to hide all non-document windows quickly, to get the maximum viewing area for the document you are working on.

8.12.7 Currently open window list

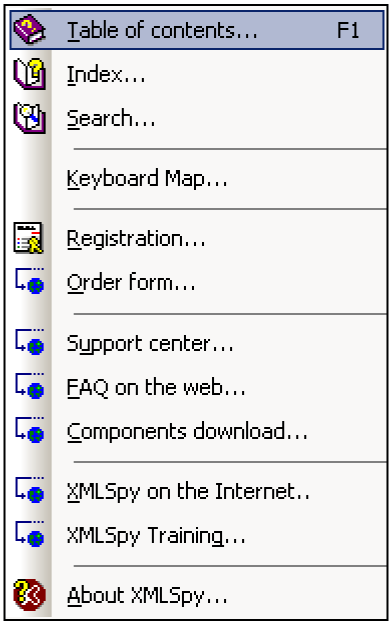
This list shows all currently open windows, and lets you quickly switch between them.



You can also use the Ctrl-TAB or CTRL F6 keyboard shortcuts to cycle through the open windows.

8.13 Help Menu

The **Help** menu contains all commands required to get help or more information on XMLSpy 2005, as well as links to information and support pages on our web server.



The **Help** menu also contains the Registration dialog, which lets you enter your license key-code, once you have purchased the product.

8.13.1 Table of contents...

This command displays a **hierarchical representation** of all chapters and topics contained in the online help system. Use this command to jump to the table of contents directly from within XMLSpy 2005.

Once the help window is open, use the three tabs to toggle between the table of contents, index, and search panes. The Favorites tab lets you bookmark certain pages within the help system.

8.13.2 Index...

This command accesses the **keyword index** of the Online Help. You can also use the Index tab in the left pane of the online help system.

The index, lists all relevant keywords and lets you navigate to a topic by double-clicking the respective keyword. If more than one topic matches the selected keyword, you are presented a list of available topics to choose from.

8.13.3 Search...

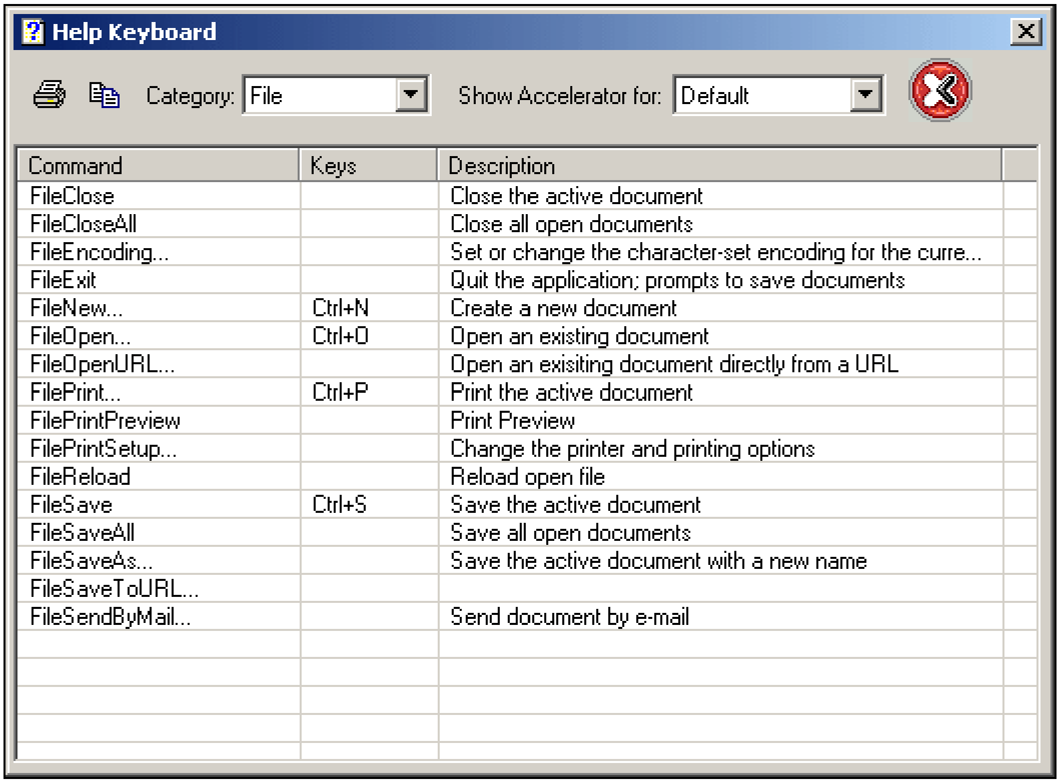
The Search command performs a **full-text search** in the entire online help system.

- Once you enter your search term into the query field and hit the Return key.
  The online help system displays a list of available topics that contain the search term you've entered.

- Double-click on any item in the list to display the corresponding topic.

8.13.4 Keyboard Map...

The **Key Map...** command causes an information box to be displayed that contains a menu-by-menu listing of all commands in XMLSpy 2005. Menu commands are listed with a description and shortcut keystrokes for the command.



Category: File

Show Accelerator for: Default

Command	Keys	Description
FileClose		Close the active document
FileCloseAll		Close all open documents
FileEncoding...		Set or change the character-set encoding for the curre...
FileExit		Quit the application; prompts to save documents
FileNew...	Ctrl+N	Create a new document
FileOpen...	Ctrl+O	Open an existing document
FileOpenURL...		Open an exisiting document directly from a URL
FilePrint...	Ctrl+P	Print the active document
FilePrintPreview		Print Preview
FilePrintSetup...		Change the printer and printing options
FileReload		Reload open file
FileSave	Ctrl+S	Save the active document
FileSaveAll		Save all open documents
FileSaveAs...		Save the active document with a new name
FileSaveToURL...		
FileSendByMail...		Send document by e-mail

To view commands in a particular menu, select the menu name in the Category combo box. You can print the command by clicking the printer icon.

8.13.5 Registration...

When you start XMLSpy 2005 for the first time, you are automatically presented with the Registration dialog box, which lets you register your software product in order to be eligible for technical support and activate your license, which is done by entering a unique key-code to unlock the software.

FREE Evaluation Version

If you have downloaded the XMLSpy 2005 from our web server and would like to activate your **FREE** 30-day evaluation version, please enter your name, company, and e-mail address and click on the "Request FREE evaluation key..." button. XMLSpy 2005 then uses your Internet connection to transmit the information you have just entered to our web server, where a personal unique evaluation license will be generated for you. The license key-code, which is necessary to unlock your software, will then be sent to the e-mail address you have entered - it is therefore important, that you enter your **real e-mail address** in the registration dialog box!

Once you have clicked the request button, please go to your favorite mail software and retrieve the license key-code from our e-mail message, which you should be receiving in a matter of a few minutes (depending on transient Internet conditions).

If you requested a key-code and it didn't arrive in a short space of time, the process may have failed due to Firewall restrictions in your network. If this is the case, please send a short message with your information via e-mail to our website and our support staff will generate a key-code for you manually.

When you have received your evaluation key-code, please enter it into the key-code field in the registration dialog box and click on OK to start working with XMLSpy 2005.

Whenever you want to place an order for a licensed version of XMLSpy 2005, you can also use the "Order license key..." button in the registration dialog box or the Order form menu command to proceed to the Secure Online Shop on the Internet.

Licensed Version

If you have purchased a *single-user* license for XMLSpy 2005, you will receive an e-mail message from us that contains your license-data and includes your name, company and key-code. Please make sure that you enter **all fields** from your license e-mail into the registration dialog box. The key-code will only be able to unlock your software installation, if the entries in the name and company fields match the name and company entered into our order form.

If your company has purchased a *multi-user* license for XMLSpy 2005, you will receive an e-mail message from us that contains your license-data and includes your company name and key-code.

Please make sure that you enter the company name and key-code from your license e-mail into the registration dialog box and also enter your personal name into the name field. The key-code will only be able to unlock your software installation, if the value in the company field match the company name entered into our order form.

Please note that the XMLSpy 2005 License-Agreement does not allow you to install more than the licensed number of copies of XMLSpy 2005 on the computers in your organization (per-seat license).

8.13.6 Order form...

When you want to place an order for a licensed version of XMLSpy 2005, use this command or the "Order license key..." button in the registration dialog to proceed to the Secure Online Shop on the Internet, where you can choose between different single- and multi-user license packs.

Once you have placed your order, you can choose to pay by credit card, send a check by mail, or use a bank wire transfer.

8.13.7 Support Center...

If you have any questions regarding our product, please feel free to use this command to send a query to the Altova Support Center at any time. This is the place where you'll find links to the FAQ, support form, and e-mail addresses for contacting our support staff directly.

8.13.8 FAQ on the web

To help you in getting the best support possible, we are providing a list of Frequently Asked Questions (FAQ) on the Internet, that is constantly updated as our support staff encounters new issues that are raised by our customers.

Please make sure to check the FAQ before contacting our technical support team. This will allow you to get help more quickly.

We regret that we are not able to offer technical support by phone at this time, but our support staff will typically answer your e-mail incidents within one business day.

If you would like to make a feature suggestion for a future version of XMLSpy 2005 or if you wish to send us any other general feedback, please use the questionnaire form.

8.13.9 Components download

The Components download option, currently lets you to download the latest Microsoft XML Parser, as well as an alternate XSLT Transformation System, and will be expanded in the future.

8.13.10 On the Internet...

This command takes you directly to the Altova web-server http://www.altova.com where you can find out about news, product updates and additional offers from the Altova team.

8.13.11 Training...

This command takes you directly to the Altova web-server http://www.altova.com where you can find out about our authorized Training Partners who provide courses on using XMLSpy 2005 and Advanced XML Application Development (AXAD).

8.13.12 About...

This command shows the XMLSpy 2005 splash screen and copyright information dialog box, which includes the XMLSpy 2005 logo.

Please note that this dialog box shows the version number - to find the number of the actual build you are using, please look at the status bar, which always includes the full version and build number.

8.14 Altova XSLT 1.0 Engine

The Altova XSLT 1.0 Engine conforms to the XSLT 1.0 Recommendation of 16 November 1999. The following limitations apply.

Whitespace in XML Document

By default, the Altova XSLT 1.0 Engine strips all whitespace in whitespace-only nodes from the source XML document. Note that the presence and absence of whitespace-only nodes affects the value the `position()` function returns.

XSLT 1.0 Elements Support

Given below is a list of XSLT 1.0 elements that have limited support.

Elements	**Support limitation**
`xsl:strip-space`	Not supported.
`xsl:preserve-space`	Not supported.

8.15 Altova XSLT 2.0 Engine

The Altova XSLT 2.0 Engine conforms to the XSLT 2.0 Working Draft of 11 February 2005. The following limitations apply.

Namespaces

Your XSLT 2.0 stylesheet should declare the following namespaces in order for you to be able to use the type constructors and functions available in XSLT 2.0. The prefixes given below are conventionally used; you could use alternative prefixes if you wish.

Namespace Name	Prefix	Namespace URI
XML Schema types	`xs:`	`http://www.w3.org/2001/XMLSchema`
XPath 2.0 functions	`fn:`	`http://www.w3.org/2005/02/xpath-functions`
XPath Datatypes	`xdt:`	`http://www.w3.org/2005/02/xpath-datatypes`

Typically, these namespaces will be declared on the `xsl:stylesheet` or `xsl:transform` element, as shown in the following listing:

```
<xsl:stylesheet version="2.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:fn="http://www.w3.org/2005/02/xpath-functions"
   xmlns:xdt="http://www.w3.org/2005/02/xpath-datatypes">
   ...
</xsl:stylesheet>
```

The following points should be noted:

- The Altova XSLT 2.0 Engine uses the XPath 2.0 Functions namespace listed above as its default functions namespace. So you can use XPath 2.0 functions and XSLT 2.0 functions in your stylesheet without any prefix. If you explicitly declare the XPath 2.0 Functions namespace in your stylesheet, you can additionally use the prefix assigned in the declaration.
- When using type constructors and types from the XML Schema and XPath Datatypes namespaces, the prefixes used in the namespace declarations must be used.

If namespaces are wrongly assigned, an error is reported. Note, however, that some XPath 2.0 functions have the same name as schema datatypes, e.g. `fn:string` and `fn:boolean`. (Both `xs:string` and `xs:boolean` are defined). So if you use the XPath expression `string('Hello')`, it should be clear from the description above that even though your intention was to use the type constructor `xs:string`, no error is reported because the expression is evaluated as `fn:string('Hello')`.

Note about XSLT 2.0 functions: The Altova XSLT 2.0 Engine automatically resolves unprefixed XSLT 2.0 functions to the XPath 2.0 Functions namespace: `http://www.w3.org/2005/02/xpath-functions`. So there is no need to use a prefix with XSLT 2.0 functions.

XPath 2.0 Functions Support

Given below is a list of built-in XPath 2.0 functions that have limited support.

Function	Support limitation
`fn:lower-case`	Applies to Latin character set only.
`fn:normalize-unicode`	Not supported.
`fn:upper-case`	Applies to Latin character set only.

XSLT 2.0 Functions Support

Given below is a list of built-in XSLT 2.0 functions that have limited support.

Function	Support limitation
`format-date`	Limitations as for `format-dateTime`.
`format-dateTime`	Presentation modifiers and formatting tokens in the variable markers of the Picture argument are not supported, and the optional Language, Calendar, and Country arguments are not supported.
`format-time`	Limitations as for `format-dateTime`.

Schema-awareness

The Altova XSLT 2.0 Engine is not schema-aware. This has the following consequences:

- For type constructors, only the built-in XML Schema and XPath datatypes are supported; user-defined types are not supported.
- Validation against a schema is not supported.

Datatypes in XPath 2.0

If you are evaluating an XPath 2.0 expression for an XML document that references an XML Schema and is valid according to this schema, you must explicitly construct or cast datatypes that are not implicitly converted to the required datatype by an operation. In the XPath 2.0 Data Model used by the Altova XSLT 2.0 Engine, all **atomized** node values from the XML document are assigned the `xdt:untypedAtomic` datatype. The `xdt:untypedAtomic` type works well with implicit type conversions. For example, the expression `xdt:untypedAtomic("1") + 1` results in a value of 2 because the `xdt:untypedAtomic` value is implicitly promoted to `xs:double` by the addition operator. Arithmetic operators implicitly promote operands to `xs:double`. Comparison operators promote operands to `xs:string` before comparing.

In some cases, however, it is necessary to explicitly convert to the required datatype. For example, if you have two elements, `startDate` and `endDate`, that are defined as being of type `xs:date` in the XML Schema, then using the XPath 2.0 expression `endDate - startDate` will show an error. On the other hand, if you use `xs:date(endDate) - xs:date(startDate)` or `(endDate cast as xs:date) - (startDate cast as xs:date)`, the expression will correctly evaluate to a singleton sequence of type `xdt:dayTimeDuration`.

Whitespace in XML Document

By default, the Altova XSLT 2.0 Engine strips all whitespace in whitespace-only nodes from the source XML document. Note that the presence and absence of whitespace-only nodes affects the value the `position()` function returns.

XSLT 2.0 Elements Support

Given below is a list of XSLT 2.0 elements that have limited support.

Elements	Support limitation
`xsl:strip-space`	Not supported.
`xsl:preserve-space`	Not supported.

8.16 Altova XQuery 1.0 Engine

The Altova XQuery 1.0 Engine conforms to the XQuery 1.0 Working Draft of 11 February 2005. This may not be the current draft when you use the engine. So be sure to use the draft of 11 February 2005 as a reference when working with the Altova XQuery 1.0 Engine. The XQuery standard gives implementations discretion about how to implement many features. Given below is a list explaining how the Altova XQuery 1.0 Engine implements these features.

Encoding
The supported character encoding is UTF-8.

Namespaces
The following namespace URIs and their associated bindings are pre-defined.

Namespace Name	Prefix	Namespace URI
XML Schema types	`xs:`	`http://www.w3.org/2001/XMLSchema`
Schema instance	`xsi:`	`http://www.w3.org/2001/XMLSchema-instance`
XPath Datatypes	`xdt:`	`http://www.w3.org/2005/02/xpath-datatypes`
Built-in functions	`fn:`	`http://www.w3.org/2005/02/xpath-functions`
Local functions	`local:`	`http://www.w3.org/2005/02/xquery-local-functions`

The following points should be noted:

- The Altova XQuery 1.0 Engine recognizes the prefixes listed above as being bound to the corresponding namespaces.
- Since the functions namespace listed above is the default functions namespace in XQuery, the `fn:` prefix does not need to be used when built-in functions are invoked (for example: `string("Hello")`). However, it can be used without having to declare the namespace in the query prolog (for example: `fn:string("Hello")`).
- You can change the default functions namespace by declaring the `default function namespace` expression in the query prolog.
- When using types from the XML Schema and XPath Datatypes namespaces, the prefixes `xs:` and `xdt:` may be used, respectively, without having to explicitly declare the namespaces and bind these prefixes to them in the query prolog. (Examples: `xs:date` and `xdt:yearMonthDuration`.) If you wish to use some other prefixes, these must be explicitly declared for the namespaces in the query prolog. (Example: `declare namespace alt = "http://www.w3.org/2001/XMLSchema"; alt:date("2004-10-04")`.)

If namespaces are wrongly assigned, an error is reported. Note, however, that some functions have the same name as schema datatypes, e.g. `fn:string` and `fn:boolean`. (Both `xs:string` and `xs:boolean` are defined). The namespace prefix determines which one is used.

XML source document and validation
XML documents used in executing an XQuery document with the Altova XQuery 1.0 Engine must be well-formed. However, they do not need to be valid according to an XML Schema. XML documents that *are* associated with an XML Schema are **not validated** against the schema before being used by the query. This means that no validation information (including type

information) is included with the XML Infoset passed to the engine. Nodes from the XML document resolve to `xdt:untypedAtomic` when they are atomized. The `xdt:untypedAtomic` type is implicitly converted to the required type by some operators. For example, the expression `xdt:untypedAtomic("1") + 1` results in a value of 2 because the `xdt:untypedAtomic` is implicitly promoted to `xs:double` by the addition operator. Arithmetic operators implicitly promote operands to `xs:double`. Comparison operators promote operands to `xs:string` before comparing.

In some cases, however, it is necessary to explicitly convert to the required datatype by constructing the required datatype or casting to it. For example, if you have two XML elements, `startDate` and `endDate`, that are defined as being of type `xs:date` in the XML Schema, then using the XPath 2.0 expression `endDate - startDate` will report an error. On the other hand, if you use `xs:date(endDate) - xs:date(startDate)` or `(endDate cast as xs:date) - (startDate cast as xs:date)`, the expression will correctly evaluate to a singleton sequence of type `xdt:dayTimeDuration`.

The XQuery `validate` instruction is not supported.

Schema imports and user-defined types
The `import schema` instruction and user-defined types are not supported.

Static and dynamic type checking
The static analysis phase checks aspects of the query such as syntax, whether external references (e.g. for modules) exist, whether invoked functions and variables are defined, and so on. No type checking is done in the static analysis phase. If an error is detected in the static analysis phase, it is reported and the execution is stopped.

Dynamic type checking is carried out at run-time, when the query is actually executed. If a type is incompatible with the requirement of an operation, an error is reported. For example, the expression `xs:string("1") + 1` returns an error because the addition operation cannot be carried out on an operand of type `xs:string`.

Library Modules
Library modules store functions and variables so they can be reused. The Altova XQuery 1.0 Engine supports modules that are stored in **a single external XQuery file**. Such a module file must contain a `module` declaration in its prolog, which associates a target namespace. Here is an example module:

```
module namespace libns="urn:module-library";
declare variable $libns:company := "Altova";
declare function libns: webaddress() { "http://www.altova.com" };
```

All functions and variables declared in the module belong to the namespace associated with the module. The module is used by importing it into an XQuery file with the `import module` statement in the query prolog. The `import module` statement only imports functions and variables declared directly in the library module file. As follows:

```
import module namespace modlib = "urn:module-library";
if    ($modlib:company = "Altova")
then   modlib:webaddress()
else   error("No match found.")
```

External functions
External functions are not supported, i.e. in those expressions using the `external` keyword, as in:

```
declare function hoo($param as xs:integer) as xs:string external;
```

Collations

The default collation is the Unicode codepoint collation. No other collation is currently supported. Comparisons, including the `fn:max` function, are based on this collation.

Character normalization

No character normalization form is supported.

Precision of numeric types

- The `xs:integer` datatype is arbitrary-precision, i.e. it can represent any number of digits.
- The `xs:decimal` datatypes has a limit of 20 digits after the decimal point.
- The `xs:float` and `xs:double` datatypes have limited-precision of 15 digits.

XQuery Instructions Support

Given below is a list of XQuery instructions that have limited support.

Instructions	Support limitation
`import schema`	Not supported.
`MUExtensions`	Not supported. Ignored.
`Pragma`	Not supported. Ignored.
`schema-attribute`	Not supported.
`schema-element`	Not supported.
`validate`	Not supported.

XQuery Functions Support

Given below is a list of built-in XQuery functions that have limited support.

Function	Support limitation
`fn:lower-case`	Applies to Latin character set only.
`fn:normalize-unicode`	Not supported.
`fn:upper-case`	Applies to Latin character set only.

XMLSpy 2005 Home Edition

XMLSpy 2005 in Eclipse Platform

9 XMLSpy 2005 in Eclipse Platform

Eclipse 3.0 is an open source framework that integrates different types of applications delivered in form of plugins. XMLSpy 2005 for the Eclipse Platform, is an Eclipse 3.0 Plug-in that allows you to access the functionality of a previously installed XMLSpy 2005 Edition from within the Eclipse 3.0 Platform.

To successfully install the XMLSpy 2005 Plug-in for Eclipse 3.0 you need the following:

- The specific XMLSpy 2005 Edition you intend to use: Enterprise, Professional, or Home
- The Eclipse 3.0 package, as well as
- The appropriate Java Runtime Edition

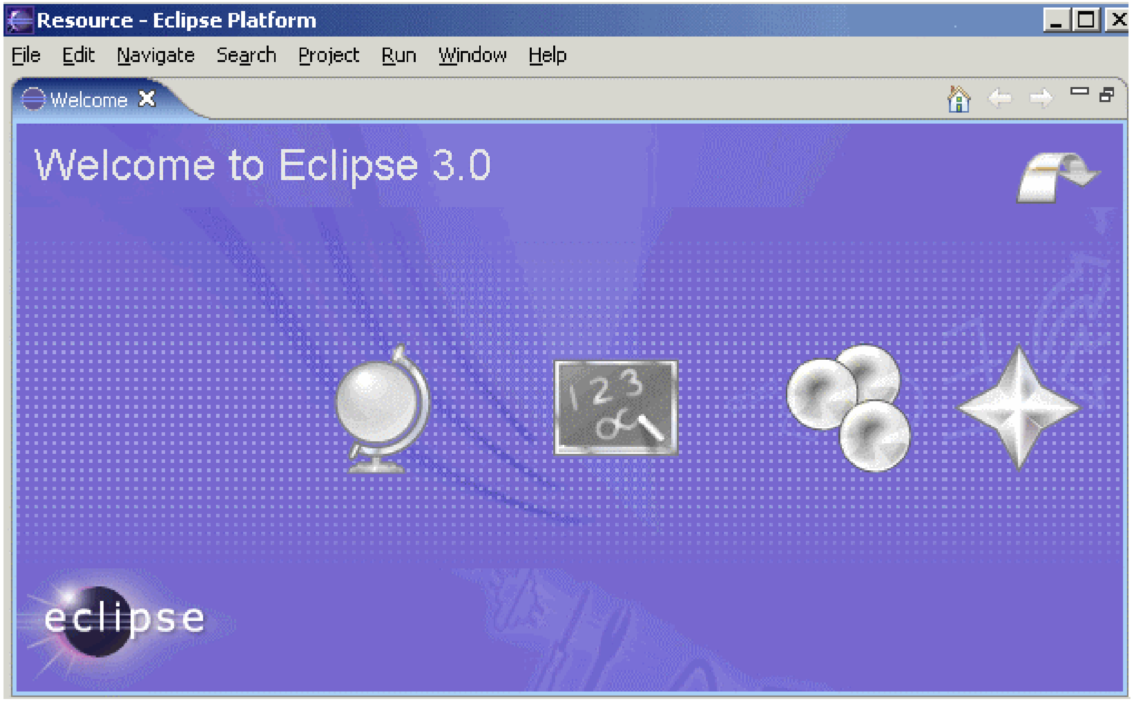
1. Download and install the XMLSpy Plugin for Eclipse from the Download section of the Altova.com website. You will be prompted for the installation folder of the Plug-in during the installation process.

The XMLSpy Plug-in for Eclipse supplies the following functionality:

- A fully-featured editor that can edit any type of file that XMLSpy 2005 is capable of editing, which also contributes application-specific actions to menu and toolbars.
- A set of Views that define the individual windows of the application: in this case the XMLSpy 2005 entry helpers.
- Different Perspectives that determine the appearance of the workbench.
- XMLSpy 2005 user help under the menu item **Help | XMLSPY |** Table of contents.

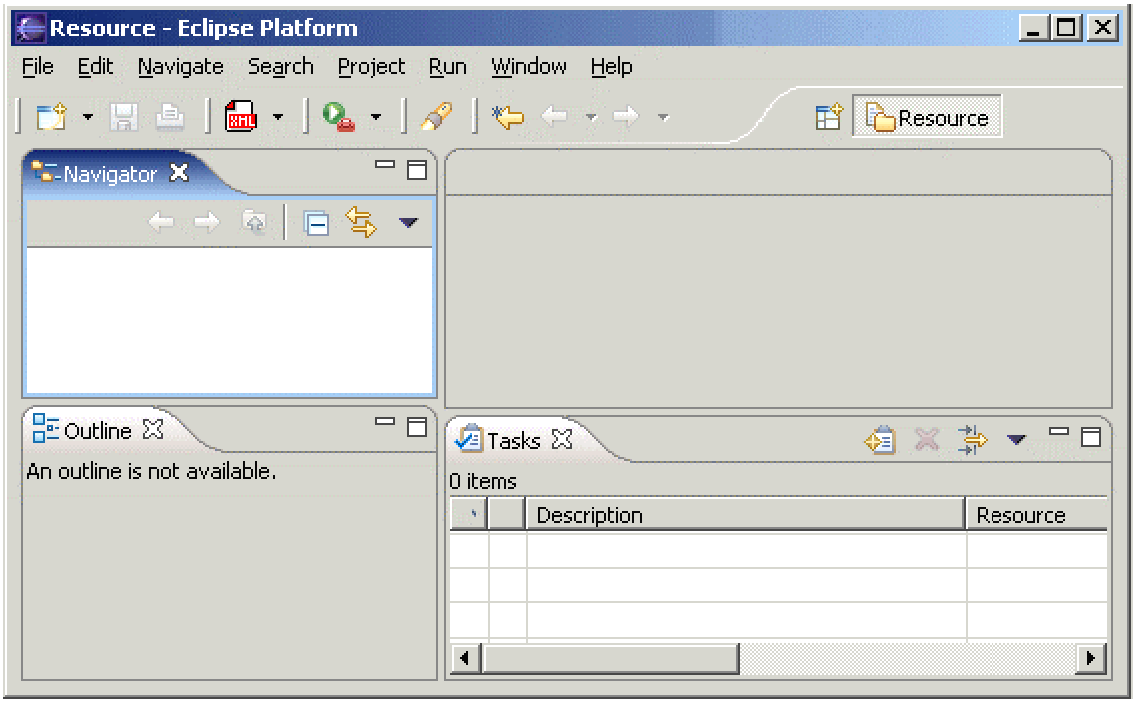
9.1 Starting Eclipse and using the XMLSpy 2005 Plug-in

1. Double click eclipse.exe to start the Eclipse Platform.


Resource - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Welcome
Welcome to Eclipse 3.0
eclipse

This opens the "Welcome to Eclipse 3.0" start screen.

2. Place the cursor over the arrow symbol, and click when the "Go To Workbench" text appears.
3. This opens an empty XMLSpy 2005 window in Eclipse.

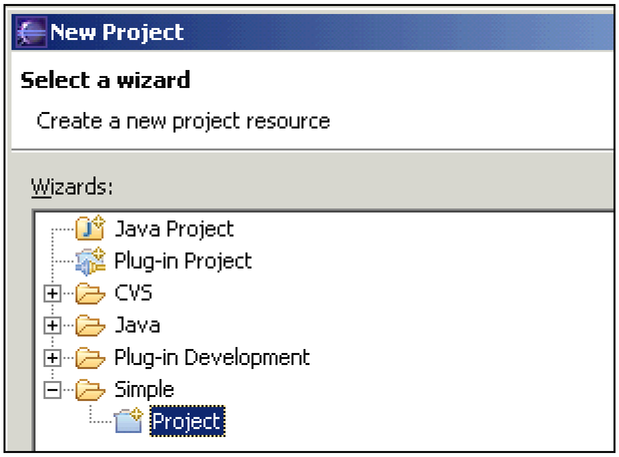

Resource - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Resource
Navigator
Outline
Tasks
Description
Resource

An outline is not available.

0 items

Creating a new Project:

1. Right click in the Navigator window, and select **New | Project | Simple Project**.

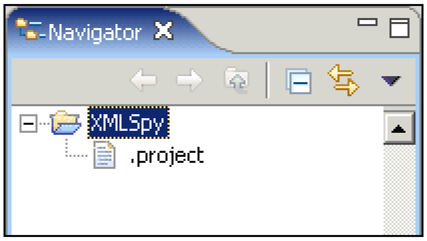


New Project

Select a wizard

Create a new project resource

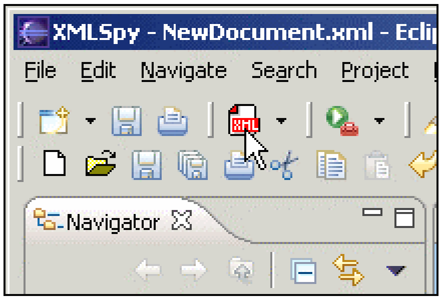
2. Enter XMLSpy as the project name, and click Finish.
3. This creates the XMLSpy project folder.



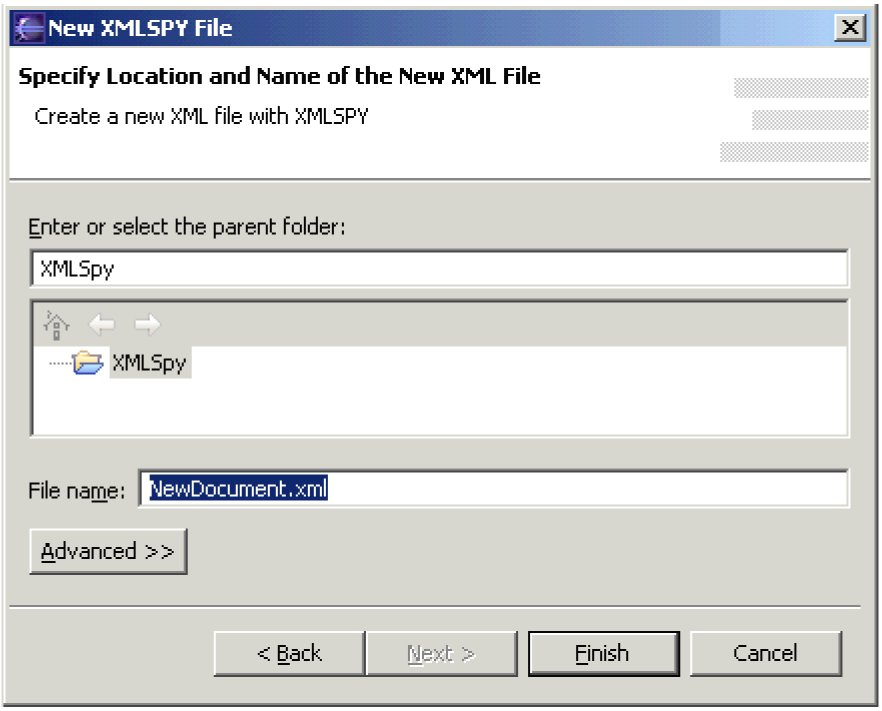
9.2 Creating XML files

Creating a new XML file based on schema

1. Click the XMLSpy 2005 XML icon.


XMLSpy - NewDocument.xml - Ecli
File Edit Navigate Search Project
XML
Navigator

2. Select **xml | XML document** then click Next.
   You are then prompted to select a parent folder.



Specify Location and Name of the New XML File

Create a new XML file with XMLSPY

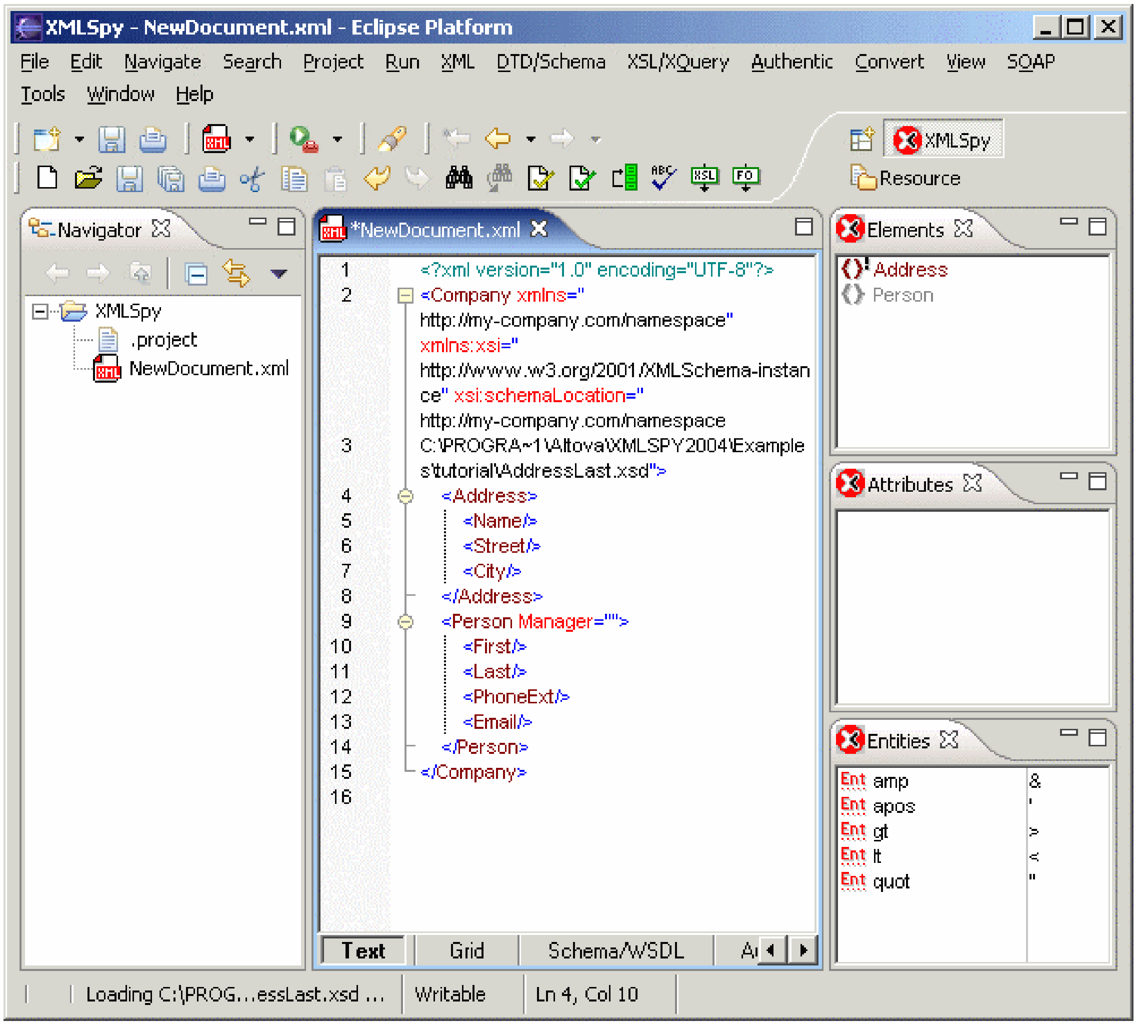
Enter or select the parent folder: XMLSpy

XMLSpy

File name: NewDocument.xml

3. Enter XMLSpy as the parent folder (or select an existing folder) and use the supplied default filename **NewDocument.xml** and click Finish.
   The dialog boxes that now appear are from XMLSpy 2005.
4. Select Schema and click OK, then select the schema file using the Browse button (e.g. AddressLast.xsd) and click OK.
   The new XML file appears in the NewDocument.xml tab in the Text view.

   The preconfigured XMLSpy 2005 perspective is automatically activated to display the various entry helpers.



```
<?xml version="1.0" encoding="UTF-8"?>
<Company xmlns="
http://my-company.com/namespace"
xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instan
ce" xsi:schemaLocation="
http://my-company.com/namespace
C:\PROGRA~1\Altova\XMLSPY2004\Example
s\tutorial\AddressLast.xsd">
  <Address>
    <Name/>
    <Street/>
    <City/>
  </Address>
  <Person Manager="">
    <First/>
    <Last/>
    <PhoneExt/>
    <Email/>
  </Person>
</Company>
```

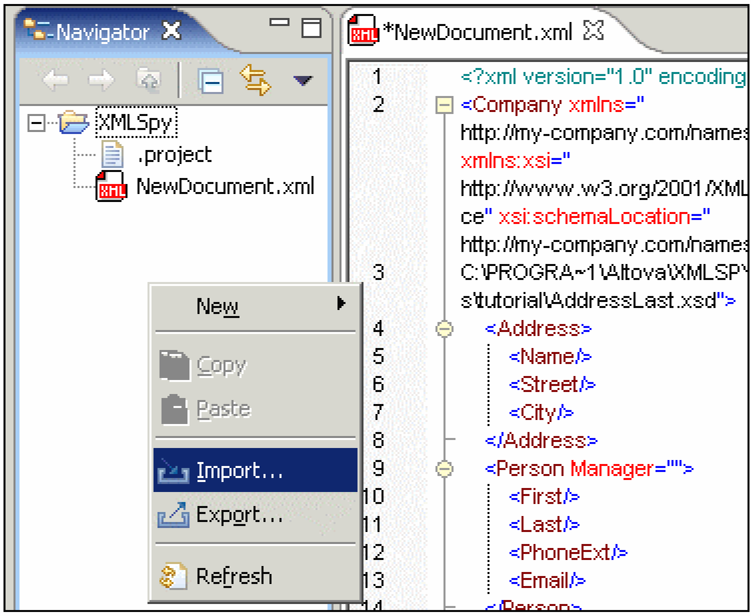
Ent amp	&
Ent apos	'
Ent gt	>
Ent lt	<
Ent quot	"

```
Loading C:\PROG...essLast.xsd ... | Writable | Ln 4, Col 10
```

9.3 Importing XML files into folders

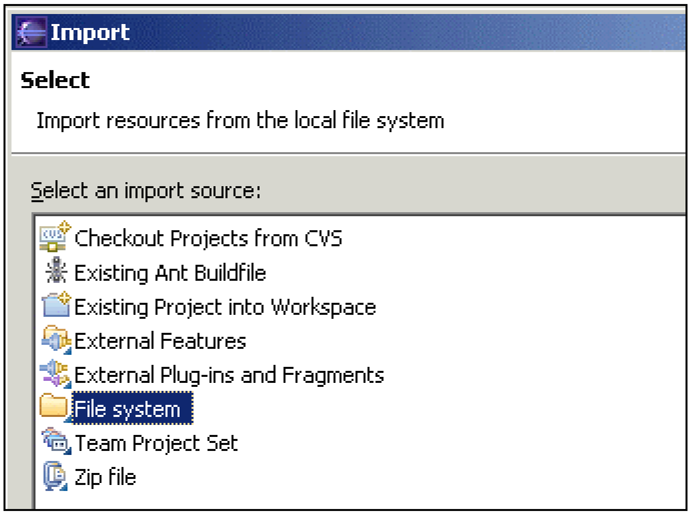
Importing XMLSpy 2005 Examples folder into the Navigator:

1. Right click the Navigator tab and click Import.



```
<?xml version="1.0" encoding
<Company xmlns="
http://my-company.com/names
xmlns:xsi="
http://www.w3.org/2001/XML
ce" xsi:schemaLocation="
http://my-company.com/names
C:\PROGRA~1\Altova\XMLSPY
s\tutorial\AddressLast.xsd">
  <Address>
    <Name/>
    <Street/>
    <City/>
  </Address>
  <Person Manager="">
    <First/>
    <Last/>
    <PhoneExt/>
    <Email/>
  </Person>
```

2. Select "File system", then click Next.



Import

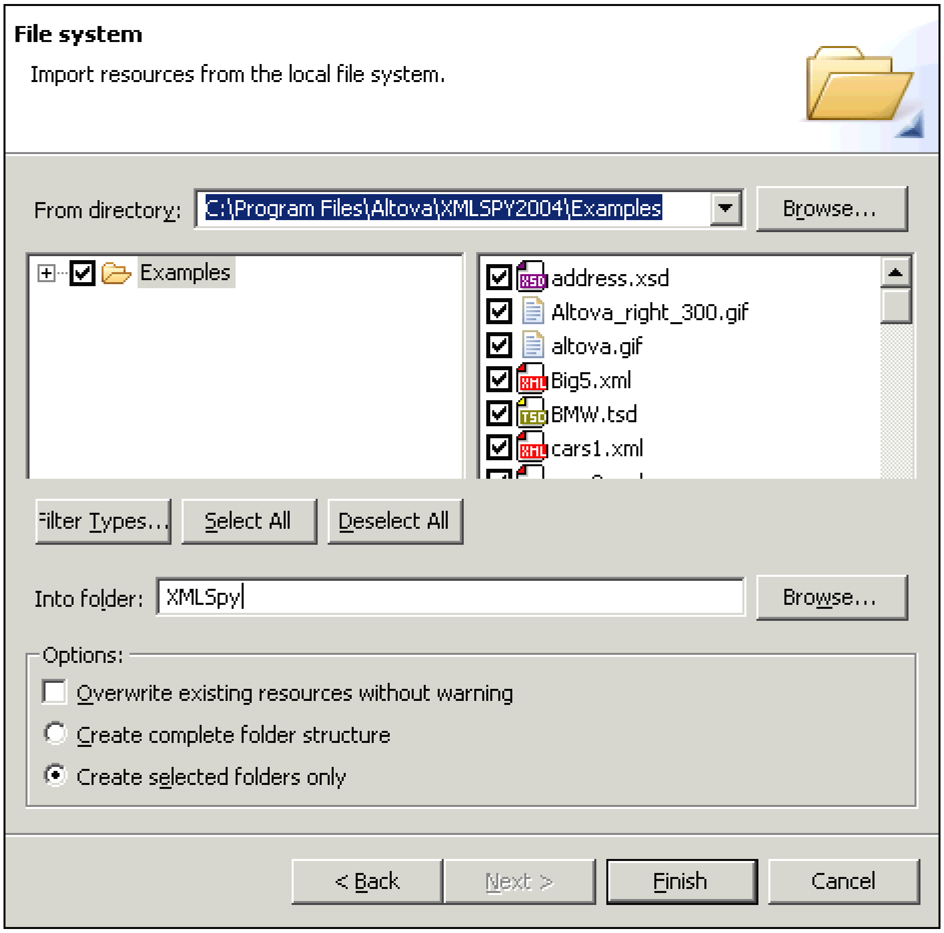
Select

Import resources from the local file system

Select an import source:

- Checkout Projects from CVS
- Existing Ant Buildfile
- Existing Project into Workspace
- External Features
- External Plug-ins and Fragments
- File system
- Team Project Set
- Zip file

3. Click the **Browse** button to the right of the "From directory:" text box, and select the Examples directory in your XMLSpy 2005 folder.



File system

Import resources from the local file system.

From directory: C:\Program Files\Altova\XMLSPY2004\Examples

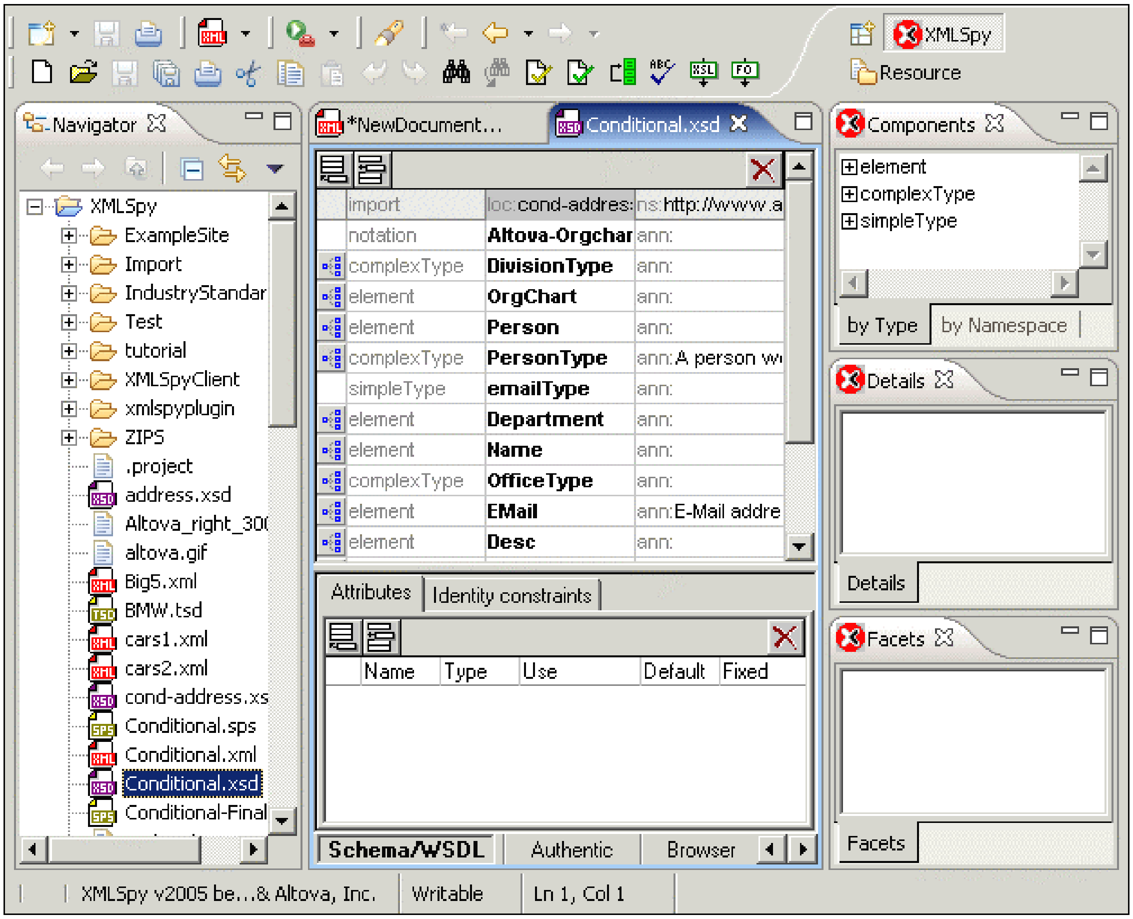
☑ Examples

☑ address.xsd
☑ Altova_right_300.gif
☑ altova.gif
☑ Big5.xml
☑ BMW.tsd
☑ cars1.xml

Into folder: XMLSpy

Options:
□ Overwrite existing resources without warning
○ Create complete folder structure
● Create selected folders only

4. Activate the **Examples** check box.
   This activates all files in the various subdirectories in the window at right.
5. Click the **Browse** button, next to the "Into folder:" text box, to select the target folder, then click Finish.
   The selected folder structure and files will be copied into the Eclipse workspace.
6. Double click a file in Navigator to open it (e.g. Conditional.xsd).



import	loc:cond-addres	ns:http://www.a
notation	**Altova-Orgchar**	ann:
complexType	**DivisionType**	ann:
element	**OrgChart**	ann:
element	**Person**	ann:
complexType	**PersonType**	ann:A person w
simpleType	**emailType**	ann:
element	**Department**	ann:
element	**Name**	ann:
complexType	**OfficeType**	ann:
element	**EMail**	ann:E-Mail addre
element	**Desc**	ann:

Name	Type	Use	Default	Fixed

XMLSpy v2005 be...& Altova, Inc. | Writable | Ln 1, Col 1

9.4 Differences between Eclipse and standalone versions

The Enterprise, Professional and Home editions of the Eclipse Plug-in for XMLSpy 2005 generally have the same functionality as their standalone counterparts.

Unsupported features in the integrated version:

Info window
The Info window is not supported. This window gives extra information on currently selected elements/attributes etc. E.g. the name of the element or attribute, the datatype, enumerations and occurrence.

Separate Browser window
The "Show in separate window by default" check box in the menu **Tools | Options | View** tab, is not supported. This means that the Text and Browser view are always incorporated in the same window when you transform an XML file to HTML file for example.

Authentic view
Text state icons are not available in the Authentic view. The functions or formatting that they may provide, are still available in the entry helper or context menu, however. Please see the Stylesheet Designer documentation for more information.

9.5 Eclipse views and perspectives

The XMLSpy 2005 Views define the main entry helpers as well as the debugger windows for each specific debugger.

- Select the menu option **Window | Show View | Other...** to display the currently available views.

The XMLSpy 2005 Perspectives define the main application window as well as the debugger user interfaces.

- Select the menu option **Window | Open Perspective| Other...** to display the currently available perspectives.

XMLSpy 2005 Home Edition

Appendices

10 Appendices

The appendices contain technical background information and licensing and distribution information relevant to your product. Information on the following topics is given in this section:

Technical Background Information

- XML Parser
- Schema dialects
- OS and memory requirements
- Internet usage
- Unicode support
- License metering

Licensing and Distribution

- Electronic software distribution
- Copyright
- Software product license

10.1 Technical Background Information

This chapter contains useful background information on the technical aspects of XMLSpy 2005. If you run across a technical term in the remainder of this help system you may occasionally find a link to some of the background materials provided here.

10.1.1 XML Parser

When opening any XML document, XMLSpy 2005 uses its built-in incremental validating parser to both check the document for well-formedness and validate it against any specified DTD, DCD, BizTalk, or XSD Schema.

The same parser is also used while editing a document that refers to a DTD, DCD, BizTalk, or XSD Schema to provide intelligent editing help and immediately display any validation error that is encountered.

This is possible through the incremental design of the new parser that is optimized for the special needs of an integrated development environment.

The built-in parser implements the Final Recommendation XML Schema from the W3C and we are constantly tracking the W3C Schema Group's efforts and are actively participating in all Schema-related discussions to provide you with a state-of-the-art development environment.

10.1.2 Schema Dialects

An important aspect of XML is the area of schemas and DTDs that define the logical structure (or content model) of an XML document - XMLSpy 2005 is the ideal tool that integrates schema and DTD creation while working with XML instance documents.

XMLSpy 2005 supports both editing and schema-validation of the following schema kinds:

- Document Type Definitions (DTD)
- Document Content Descriptions (DCD)
- BizTalk
- XML Schema Definition (XSD) draft April 7, 2000, CR Oct. 24 2000, May 2nd 2001 Final Recommendation

and can validate an XML instance document against any of the above schema dialects.

10.1.3 Altova XSLT and XQuery Engines

Altova products use the Altova XSLT 1.0 Engine, Altova XSLT 2.0 Engine, and Altova XQuery Engines. Documentation about implementation-specific behavior for these engines is in the User Reference section of the product documentation where relevant.

These three engines are also available as standalone applications that can be downloaded from the Altova website free of charge. Documentation for using the engines is available with the application package.

10.1.4 OS & Memory Requirements

XMLSpy 2005 is a modern 32-bit Windows application that runs on Windows NT 4.0, Windows 2000 and Windows XP. It requires a fair amount of memory to be installed in the system, because it loads each document fully into memory.

On the other hand it typically requires less memory than many Java-based applications, because it is written entirely in C++ and thus does not require the overhead of a Java runtime environment.

Having documents in main memory is necessary to completely parse and analyze each document, and to also improve the viewing and editing speed during normal work. While editing a small to medium sized document (up to 512kB) is possible in as little as 2MB of RAM, opening a 5MB document can consume up to 50MB during the initial parsing process.

Memory requirements are also influenced by the unlimited Undo history. When repeatedly cutting and pasting large selections in large documents, memory can rapidly be depleted.

10.1.5 Internet Usage

XMLSpy 2005 is an integrated development environment for XML and as such will, also initiate Internet connections on your behalf in the following situations:

- If you click the "Request evaluation key-code" in the registration dialog the three fields in the registration dialog box are transferred to our web server by means of a regular http (port 80) connection and the free evaluation key-code is sent back to the customer via regular SMTP e-mail.
- If you use the Open URL... dialog box to open a document directly from a URL, that document is retrieved through a http (port 80) connection.
- If you open an XML document that refers to an XML Schema or DTD and the document is specified through a URL, it is also retrieved through a http (port 80) connection, once you validate the XML document. This may also happen automatically upon opening a document, if you have instructed XMLSpy 2005 to automatically validate files upon opening in the File tab of the Tools | Options dialog.
- If you are using the Send by mail... command, the current selection or file is sent by means of any MAPI-compliant mail program installed on the user's PC.

All this communication is, of course, only initiated in response to a direct request from you! XML is, after all, related to the Internet and thus any XML development tool must have access Internet protocols to provide an efficient environment for the everyday duties of any XML developer.

10.1.6 Unicode Support

Unicode is the new 16-bit character-set standard defined by the Unicode Consortium that provides a unique number for every character,

- no matter what the platform,
- no matter what the program,
- no matter what the language.

Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one. Before Unicode was invented, there were hundreds of different encoding systems for assigning these numbers. No single encoding could contain enough characters: for example, the European Union alone requires several different encodings to cover all its languages. Even for a single language like English, no single encoding was adequate for all the letters, punctuation, and technical symbols in common use.

These encoding systems used to conflict with one another. That is, two encodings used the same number for two different characters, or different numbers for the same character. Any given computer (especially servers) needs to support many different encodings; yet whenever data is passed between different encodings or platforms, that data always runs the risk of corruption.

Unicode is changing all that!
Unicode provides a unique number for every character, no matter what the platform, no matter what the program, and no matter what the language. The Unicode Standard has been adopted by such industry leaders as Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Base and many others.

Unicode is required by modern standards such as XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc., and is the official way to implement ISO/IEC 10646. It is supported in many operating systems, all modern browsers, and many other products. The emergence of the Unicode Standard, and the availability of tools supporting it, are among the most significant recent global software technology trends.

Incorporating Unicode into client-server or multi-tiered applications and web sites offers significant cost savings over the use of legacy character sets. Unicode enables a single software product or a single web site to be targeted across multiple platforms, languages and countries without re-engineering. It allows data to be transported through many different systems without corruption.

Even though XML is clearly defined to be based on the Unicode standard, XMLSpy 2005 is still one of the few XML development tools that *fully* implements Unicode!

Windows NT4.0/2000/XP

Starting with version 2.0 XMLSpy 2005 provided full Unicode support in the Windows NT, Windows 2000, and Windows XP versions of the software. To edit any XML document from a non-roman writing system you will, however, also need a font that supports the Unicode characters being used by that document.

Windows NT typically includes support for all common single-byte writing-systems in its Arial, Times, and Courier New fonts and will additionally include all required fonts for the writing-system in your own country (i.e. if you install the Japanese version of Windows NT you will automatically have fonts that support the Katakana, Hiragana, and Kanji writing-systems as well as the input-methods and dictionaries to enter Kanji and to switch between Katakana and Hiragana). If you wish to edit any document from a foreign writing-system, you may want to install additional Windows NT components for that writing-system or purchase special Unicode fonts for these writing-systems (such fonts are available from all leading type vendors).

Please note that most fonts only contain a very specific subset of the entire Unicode range and are therefore typically targeted at the corresponding writing system. Consequently you may encounter XML documents that contain "unprintable" characters, because the font you have selected does not contain the required glyphs. Therefore it can sometimes be very useful to have a font that covers the entire Unicode range - especially when editing XML documents from all over the world.

The most universal font we have encountered is a typeface called "Arial Unicode MS" that has been created by Agfa Monotype for Microsoft. This font contains over 50.000 glyphs and covers the entire set of characters specified by the Unicode 2.1 standard. It needs 23MB and is included with Microsoft Office 2000.

We highly recommend that you install this font on your system and use it with XMLSpy 2005, if you are often editing documents in different writing systems. This font is not installed with the "typical" setting of the Microsoft Office setup program, but you can choose the Custom Setup option to install this font.

In the "Examples" folder you will also find a new XHTML file called `Unicode-UTF8.html` that contains the sentence "When the world wants to talk, it speaks Unicode" in many different languages ("Wenn die Welt miteinander spricht, spricht sie Unicode") and writing-systems (世界的に話すなら、Unicode です。) - this line has been adopted from the 10th Unicode conference in 1997 and is a beautiful illustration of the importance of Unicode for the XML standard. Opening this file will give you a quick impression on what is possible with Unicode and what writing systems are supported by the fonts available on your PC installation.

Right-to-Left Writing Systems

Please note that even under Windows NT 4.0 any text from a right-to-left writing-system (such as Hebrew or Arabic) is not rendered correctly except in those countries that actually use right-to-left writing-systems. This is due to the fact that only the Hebrew and Arabic versions of Windows NT contains support for rendering and editing right-to-left text on the operating system layer.

10.1.7 License Metering

XMLSpy 2005 has a built-in license metering module that helps you in avoiding any unintentional violation of our license agreement. XMLSpy 2005 can be licensed either as a single-user or multi-user software and depending on your license, this license-metering module makes sure, that no more than the licensed number of users are using XMLSpy 2005 concurrently.

This license-metering technology uses your local area network (LAN) to communicate between instances of XMLSpy 2005 running on different computers.

Single license
When XMLSpy 2005 starts up, it sends a short broadcast datagram to find any other instance of the product running on another computer in the same network segment. If it doesn't get any response, it will open a port for listening to other instances of XMLSpy 2005. Other than that, it will do nothing at all in a single-user situation. If you are not connected to a LAN or are using dial-up connections to connect to the Internet, XMLSpy 2005 will NOT generate any network traffic at all.

Multi license
If more than one copy of XMLSpy 2005 is used within the same LAN, they will briefly communicate with each other on startup, to exchange their key-codes to ensure that the number of concurrent licenses purchased is not accidentally violated, as additional copies of the product are launched by more users.

This is the same kind of license metering technology, that is common in the Unix world and with many other database development tools and allows our customers to purchase reasonably-priced concurrent-use multi-user licenses (see http://www.altova.com/order for our price list).

Please note, that XMLSpy 2005 is at no time attempting to send any information out of your LAN, or over the Internet. We are also deliberately sending very few and small network packets so as to not put a burden on any network. The TCP/IP ports (2799) used by XMLSpy 2005 are officially registered with the IANA see http://www.isi.edu/in-notes/iana/assignments/port-numbers for details) and our license-metering module is a proven and tested technology.

If you are using a firewall, you may notice communications on port 2799 between the computers that are running XMLSpy 2005. You are, of course, free to block such traffic between different groups in your organization, as long as you can ensure by other means, that your license agreement is not violated.

You will also notice, that XMLSpy 2005 contains many useful functions that make use of your Internet connection, but these are unrelated to the license-metering technology.

10.2 Licensing and Distribution

This appendix contains important legal information concerning your rights to use this software product. Please read carefully - this information is binding, as you have agreed to these terms upon installation of this software product.

10.2.1 Electronic Software Distribution

We are making XMLSpy 2005 available through electronic software distribution only, because this method offers many unique benefits for our customers:

- you can freely evaluate the software before making a purchasing decision
- once you decide to buy the software, you can place your order online and immediately get a fully licensed product within minutes
- you can be sure to always get the latest version of our software
- we include both a comprehensive integrated online-help system and an electronic manual that you can also print out, if you prefer to read your documentation on paper

Once you download this software product, you may evaluate the XMLSpy 2005 for a period of up to 30 days free of charge. During this evaluation period the software will start to remind you after about 20 days that it has not been licensed yet. The reminder message will, however, only be displayed once every time you start the program.

If you would like to continue using the program after the 30 day evaluation period, you have to purchase a SOFTWARE PRODUCT LICENSE, which is delivered in the form of a key-code that you enter into the Registration dialog to unlock the product.

You can register and purchase your license on-line by directing your browser to access our web-shop at http://www.altova.com/order. On this page you will get detailed pricing information (including multi-user discounts) and also find a list of authorized distributors and resellers.

If you want to share XMLSpy 2005 with others, please make sure that only the installation program is ever distributed. It contains the application program, grammar description, sample files, and this online manual as well as a quick Read-Me file in one neat package. Any person that receives the XMLSpy 2005 software from you is also automatically entitled to a 30 day evaluation period. After the expiration of said period, any other user must also purchase a license in order to be able to use XMLSpy 2005.

For further details, please refer to the SOFTWARE PRODUCT LICENSE at the end of this manual.

10.2.2 Copyright

All title and copyrights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT are owned by Altova GmbH or its suppliers. The SOFTWARE PRODUCT is protected by copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE PRODUCT like any other copyrighted material.

Copyright ©1998-2005 Altova GmbH

All rights reserved.

The Sentry Spelling-Checker Engine Copyright © 2000 Wintertree Software Inc.

STLport Copyright 1999,2000 Boris Fomitchev, Copyright 1994 Hewlett-Packard Company, Copyright 1996,97 Silicon Graphics Computer Systems, Inc., Copyright 1997 Moscow Center for SPARC Technology.

10.2.3 Software Product License

THIS IS A LEGAL DOCUMENT -- RETAIN FOR YOUR RECORDS

ALTOVA® END USER LICENSE AGREEMENT
FOR XMLSPY HOME EDITION SOFTWARE

Licensor:

Altova GmbH
Rudolfsplatz 13a/9
A-1010 Wien
Austria

Important - Read Carefully. Notice to User:

This End User License Agreement FOR XMLSPY HOME-EDITION SOFTWARE ("XMLSPYHOME-EULA"） governs your right to use the XMLSPY Home-Edition Software ("Home-Edition Software"). This XMLSPYHOME-EULA is a legal document between you and Altova GmbH ("Altova"). It is important that you read this document before using the Altova-provided software and any accompanying documentation, including, without limitation printed materials, 'online' files , or electronic documentation ("Documentation"). By clicking the "I accept" and "Next" buttons below, or by installing, or otherwise using the Home-Edition Software, you agree to be bound by the terms of this XMLSPYHOME-EULA as well as the Altova Privacy Policy ("Privacy Policy") including, without limitation, the warranty disclaimers, limitation of liability, data use and termination provisions below. You agree that this agreement is enforceable like any written agreement negotiated and signed by you. If you do not agree, you are not licensed to use the Home-Edition Software, and you must destroy any downloaded copies of the Home-Edition Software in your possession or control. Please go to our Web site at http://www.altova.com/xmlspyhomeditioneula to download and print a copy of this XMLSPYHOME-EULA for your files and http://www.altova.com/privacy to review the privacy policy.

1. XMLSPY HOME- EDITION SOFTWARE LICENSE

(a) **License Grant For Internal Use.** Upon your acceptance of this XMLSPYHOME-EULA Altova grants you a personal non-exclusive, **non-transferable**, limited license to install and use a copy of the Home-Edition Software on your compatible computer. You may install one copy of such Setup Program for the Home-Edition Software on a computer file server within your internal network for the sole and exclusive purpose of installing the Home-Edition (to an unlimited number of client computers on your internal network). No other server or network use of the Home-Edition Software is permitted, including but not limited to using the (i) either directly or through commands, data or instructions from or to another computer or (ii) for internal network, internet or web hosting services.
.

(b) **Backup and Archival Copies.** You may make one backup and one archival copy of the Home-Edition Software, provided your backup and archival copies are not installed or used on any computer and further provided that all such copies shall bear the original and unmodified copyright, patent and other intellectual property markings that appear on or in the Home-Edition Software. You may not transfer the rights to a backup or archival copy unless you transfer all rights in the Home-Edition Software as provided under Section 3.

(c) **Key Codes.** Upon acceptance of this XMLSPYHOME-EULA, you will receive a key code that will enable you to activate or operate the Home-Edition Software. You may not re-license, reproduce or distribute any key code except with the express written permission of Altova.

(d) **Title.** Title to the Home-Edition Software is not transferred to you. Ownership of all copies of the Home-Edition Software and of copies made by you is vested in Altova, subject to the rights of use granted to you in this XMLSPYHOME-EULA. As between you and Altova, documents, files, stylesheets, generated program code and schemas that are authored or created by you via your utilization of the Home-Edition Software, in accordance with its Documentation and the terms of this XMLSPYHOME-EULA, are your property.

(e) **Reverse Engineering.** Except and to the limited extent as may be otherwise specifically provided by applicable law in the European Union, you may not reverse engineer, decompile, disassemble or otherwise attempt to discover the source code, underlying ideas, underlying user interface techniques or algorithms of the Home-Edition Software by any means whatsoever, directly or indirectly, or disclose any of the foregoing, except to the extent you may be expressly permitted to decompile under applicable law in the European Union, if it is essential to do so in order to achieve operability of the Home-Edition Software with another software program, and you have first requested Altova to provide the information necessary to achieve such operability and Altova has not made such information available. Altova has the right to impose reasonable conditions and to request a reasonable fee before providing such information. Any information supplied by Altova or obtained by you, as permitted hereunder, may only be used by you for the purpose described herein and may not be disclosed to any third party or used to create any Home-Edition Software which is substantially similar to the expression of the Home-Edition Software. Requests for information from users in the European Union with respect to the above should be directed to the Altova Customer Support Department.

(f) **Other Restrictions.** You may not loan, rent, lease, sublicense, distribute or otherwise transfer all or any portion of the Home-Edition. You may not copy the Home-Edition Software except as expressly set forth above, and any copies that you are permitted to make pursuant to this XMLSPYHOME-EULA must contain the same copyright, patent and other intellectual property markings that appear on or in the Home-Edition Software. You may not modify, adapt or translate the Home-Edition Software. You may not, directly or indirectly, encumber or suffer to exist any lien or security interest on the Home-Edition Software; knowingly take any action that would cause the Home-Edition Software to be placed in the public domain; or use the Home-Edition Software in any computer environment not specified in this XMLSPYHOME-EULA. You will comply with applicable law and Altova's instructions regarding the use of the Home-Edition Software. You agree to notify your employees and agents who may have access to the Home-Edition Software of the restrictions contained in this XMLSPYHOME-EULA and to ensure their compliance with these restrictions. you agree that you are solely responsible for the accuracy and adequacy of the Home-Edition Software for your intended use and you will indemnify and HOLD harmless ALTOVA from any 3rd party suit to the extent based upon the accuracy and adequacy of the Home-Edition Software in your use. without limitation, The Home-Edition Software is not intended for use in the operation of nuclear facilities, aircraft navigation, communication systems or air traffic control equipment, where the failure of the Home-Edition Software could lead to death, personal injury or severe physical or environmental damage.

2. INTELLECTUAL PROPERTY RIGHTS

Acknowledgement of Altova's Rights. You acknowledge that the Home-Edition Software and any copies that you are authorized by Altova to make are the intellectual property of and are owned by Altova and its suppliers. The structure, organization and code of the Home-Edition Software are the valuable trade secrets and confidential information of Altova and its suppliers. The Home-Edition Software is protected by copyright, including without limitation by United States Copyright Law, international treaty provisions and applicable laws in the country in which it is being used. You acknowledge that Altova retains the ownership of all patents, copyrights, trade secrets, trademarks and other intellectual property rights pertaining to the Home-Edition Software, and that Altova's ownership rights extend to any images, photographs, animations, videos, audio, music, text and "applets" incorporated into the Home-Edition Software and all accompanying printed materials. You will take no actions which adversely affect Altova's intellectual property rights in the Home-Edition Software. Trademarks shall be used in accordance with accepted trademark practice, including identification of trademark owners' names. Trademarks may only be used to identify printed output produced by the Home-Edition Software, and such use of any trademark does not give you any right of ownership in that trademark. XMLSPY, AUTHENTIC, STYLEVISION, MAPFORCE, MARKUP YOUR MIND, AXAD, NANONULL, and ALTOVA are trademarks of Altova GmbH (registered in numerous countries). Unicode and the Unicode Logo are trademarks of Unicode, Inc. Windows, Windows 95, Windows 98, Windows NT, Windows 2000 and Windows XP are trademarks of Microsoft. W3C, CSS, DOM, MathML, RDF, XHTML, XML and XSL are trademarks (registered in numerous countries) of the World Wide Web Consortium (W3C); marks of the W3C are registered and held by its host institutions, MIT, INRIA and Keio. Except as expressly stated above, this XMLSPYHOME-EULA does not grant you any intellectual property rights in the Home-Edition Software. Notifications of claimed copyright infringement should be sent to Altova's copyright agent as further provided on the Altova Web Site.

3. WARRANTY DISCLAIMER AND LIMITATION OF LIABILITY

(a) THE HOME-EDITION SOFTWARE IS PROVIDED TO YOU FREE OF CHARGE, AND ON AN "AS-IS" BASIS. ALTOVA PROVIDES NO WARRANTIES FOR THE HOME-EDITION SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, ALTOVA AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND REPRESENTATIONS, WHETHER EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE; MERCHANTABILITY; SATISFACTORY QUALITY, INFORMATIONAL CONTENT, OR ACCURACY, QUIET ENJOYMENT, TITLE, AND NON- INFRINGEMENT. ALTOVA DOES NOT WARRANT THAT THE HOME-EDITION SOFTWARE IS ERROR-FREE OR WILL OPERATE WITHOUT INTERRUPTION. IF APPLICABLE LAW REQUIRES ANY WARRANTIES WITH RESPECT TO THE HOME-EDITION SOFTWARE, ALL SUCH WARRANTIES ARE LIMITED IN DURATION TO 30 DAYS FROM THE DATE OF INSTALLATION OR USE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM STATE TO STATE. you agree that you are solely responsible for the accuracy and adequacy of the software for your intended use and you will indemnify and HOLD harmless ALTOVA from any 3rd party suit to the extent based upon the accuracy and adequacy of the software in your use. Without limitation, The Software is not intended for use in hazardous environments REQUIRING fail-safe controls INCLUDING without LIMITATION THE operation of nuclear facilities, aircraft navigation, communication systems, AIR traffic CONTROL, life SUPPORT, or weapons systems, where the failure of the Software could lead to death, personal INJURY, or severe physical or environmental damage.

(b) TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL ALTOVA OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE HOME-EDITION SOFTWARE, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, OR ANY PROVISION OF THIS XMLSPY HOME-EDITION END USER LICENSE AGREEMENT, EVEN IF ALTOVA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. WHERE LEGALLY, LIABILITY CANNOT BE EXCLUDED, BUT MAY BE LIMITED, ALTOVA'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE SUM OF FIFTY DOLLARS (USD. $50) IN TOTAL. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT

APPLY TO YOU. IN SUCH STATES AND JURISDICTIONS, ALTOVA'S LIABILITY AND THAT OF ITS SUPPLIERS SHALL BE LIMITED TO THE GREATEST EXTENT PERMITTED BY LAW. THE FOREGOING LIMITATIONS ON LIABILITY ARE INTENDED TO APPLY TO THE WARRANTIES AND DISCLAIMERS ABOVE AND ALL OTHER ASPECTS OF THIS XMLSPY HOME-EDITION END USER LICENSE AGREEMENT

4. HOME-EDITION SOFTWARE ACTIVATION, UPDATES METERING AND DATA USE

Altova has a built-in license metering module that helps you to avoid any unintentional violation of this XMLSPYHOME-EULA. Altova may use your internal network for license metering. Altova's Home-Edition Software may use your internal network and internet connection for the purpose of transmitting license-related data at the time of installation, registration or update to an Altova-operated license server and validating the authenticity of the license-related data in order to protect Altova against unlicensed or illegal use of the Home-Edition Software and to improve customer service. . Activation is based on the exchange of license related data between your computer and the Altova license server. You agree that Altova may use these measures and you agree to follow any applicable requirements. In the event that Altova provides LiveUpdate notification service(s) to you, Altova may use your internal network and internet connection for the purpose of transmitting license-related data to an Altova-operated LiveUpdate server to validate your license at appropriate intervals and determine if there is any update available for you. From time to time, Altova may offer you the option to try out advanced features of its XMLSPY Professional or Enterprise Edition products for a limited time. If that is the case, then your use of these advanced features is for evaluation purposes only and is otherwise governed by this XMLSPYHOME-EULA. Altova may use your internal network and internet connection for the purpose of transmitting license-related data to an Altova-operated licensing server to provide you with a time-limited evaluation key for any advanced feature that Altova chooses to make available to you at its sole discretion. You may only request to try one advanced feature per day, and you may only request to try the same feature for a limited number of times. Altova is free to decline any such request at its sole discretion. You agree that Altova may collect information about the features you are requesting. The terms and conditions of the Privacy Policy are set out in full at http://www.altova.com/privacy and are incorporated by reference into this XMLSPYHOME-EULA. By your acceptance of the terms of this XMLSPYHOME-EULA or use of the Home-Edition Software, you authorize the collection, use and disclosure of information collected by Altova for the purposes provided for in this XMLSPYHOME-EULA and/or the Privacy Policy as revised from time to time. European users understand and consent to the processing of personal information in the United States for the purposes described herein. Altova has the right in its sole discretion to amend this provision of the XMLSPYHOME-EULA and/or Privacy Policy at any time. You are encouraged to review the terms of the Privacy Policy as posted on the Altova website from time to time.

5. TERM AND TERMINATION

Without prejudice to any other rights or remedies of Altova, this XMLSPYHOME-EULA may be terminated (a) by you giving Altova written notice of termination; or (b) by Altova, at its option, giving you written notice of termination or (c) Altova giving you written notice of termination if you fail to comply with the terms and conditions of the XMLSPYHOME-EULA. Upon any termination of this XMLSPYHOME-EULA, you must cease all use of Home-Edition Software, licensed hereunder, destroy all copies then in your possession or control and take such other actions as Altova may reasonably request to ensure that no copies of the Home-Edition Software remain in your possession or control. The terms and condition set forth in Sections 1 (d), (e), (f), 3, 5, 7 and 8 survive termination as applicable.

6. SUPPORT

Altova provides no technical support for the version of the Home-Edition Software product edition that you have licensed unless you have elected to purchase a "Support & Maintenance Package ("SMP"). The Support Period, hereinafter defined, covered by such

SMP shall be delineated at such time as you elect to purchase a SMP. Your rights with respect to support as well as your upgrade eligibility depend on your decision to purchase a SMP and the level of SMP that you have purchased. All technical support is provided via a Web-based support form only, and there is no guaranteed response time. If you have purchased SMP, then solely for the duration of its delineated Support Period, you are eligible to receive the version of the Home-Edition Software edition that you have licensed and all maintenance releases and updates for that edition that are released during your Support Period. For the duration of your SMP's Support Period, you will also be eligible to receive upgrades to the comparable edition of the next version of the Home-Edition Software that succeeds the edition that you have licensed for applicable upgrades released during your Support Period. Software that is introduced as separate product is not included in SMP. Maintenance releases, updates and upgrades may or may not include additional features. In addition, Altova will provide Priority Technical Support to you for the duration of the Support Period. Priority Technical Support is provided via a Web-based support form only, and Altova will make commercially reasonable efforts to respond via e-mail to all requests within forty-eight (48) hours during Altova's business hours (MO-FR, 8am UTC – 10pm UTC, Austrian and US holidays excluded) and to make reasonable efforts to provide work-arounds to errors reported in the Software.

During the Support Period you may also report any Home-Edition Software problem or error to Altova. In its sole discretion, Altova may elect to correct or provide a usable work-around solution in an upcoming maintenance release or update, which is made available at certain times at Altova's sole discretion. If Altova, in its discretion, requests written verification of an error or malfunction discovered by you or requests supporting example files that exhibit the Home-Edition Software problem, you shall promptly provide such verification or files, by email, telecopy, or overnight mail, setting forth in reasonable detail the respects in which the Home-Edition Software fails to perform. You shall use reasonable efforts to cooperate in diagnosis or study of errors. Altova may include error corrections in maintenance releases, updates, or new major releases of the Home-Edition Software. Altova is not obligated to fix errors. Whether or not you have purchased the Support & Maintenance Package, technical support only covers issues or questions resulting directly out of the operation of the Home-Edition Software and Altova will not provide you with generic consultation, assistance, or advice under any circumstances. The provision of technical support in no way constitutes a warranty for Home-Edition Software or the use or performance thereof

Updating Home-Edition Software may require the updating of software not covered by this XMLSPYHOME-EULA before installation. Updates of the operating system and application software not specifically covered by this XMLSPYHOME-EULA are your responsibility and will not be provided by Altova under this XMLSPYHOME-EULA. Altova's support obligations under this Section 6 are contingent upon your proper use of the Home-Edition Software and your compliance with the terms and conditions of this XMLSPYHOME-EULA at all times. Altova shall be under no obligation to provide the above technical support if, in Altova's opinion, the Home-Edition Software has failed due to the following conditions: (i) damage caused by the relocation of the software to another location or CPU; (ii) alterations, modifications or attempts to change the Home-Edition Software without Altova's written approval; (iii) causes external to the Home-Edition Software, such as natural disasters, the failure or fluctuation of electrical power, or computer equipment failure; (iv) your failure to maintain the Home-Edition Software at Altova's specified release level; or (v) use of the Home-Edition Software with other software without Altova's prior written approval. It will be your sole responsibility to: (i) comply with all Altova-specified operating and troubleshooting procedures and then notify Altova immediately of the Home-Edition Software malfunction and provide Altova with complete information thereof; (ii) provide for the security of your confidential information; (iii) establish and maintain backup systems and procedures necessary to reconstruct lost or altered files, data or programs.

7. RESTRICTED RIGHTS NOTICE AND EXPORT RESTRICTIONS.

The Home-Edition Software was developed entirely at private expense and is commercial computer software provided with **RESTRICTED RIGHTS**. Use, duplication or disclosure by the U.S. Government or a U.S. Government contractor or subcontractor is subject to the restrictions set forth in this Agreement and as provided in FAR 12.211 and 12.212 (48 C.F.R.

§12.211 and 12.212) or DFARS 227. 7202 (48 C.F.R. §227-7202) as applicable. Consistent with the above as applicable, Commercial Computer Software and Commercial Computer Documentation is licensed to U.S. government end users only as commercial items and only with those rights as are granted to all other end users under the terms and conditions set forth in this XMLSPYHOME-EULA. Manufacturer is Altova GmbH, Rudolfsplatz, 13a/9, A-1010 Vienna, Austria/EU. You may not use or otherwise export or re-export the Home-Edition Software or Documentation except as authorized by United States law and the laws of the jurisdiction in which the Home-Edition Software was obtained. In particular, but without limitation, the Home-Edition Software or Documentation may not be exported or re-exported (i) into (or to a national or resident of) any U.S. embargoed country or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce's Table of Denial Orders. By using the Home-Edition Software, you represent and warrant that you are not located in, under control of, or a national or resident of any such country or on any such list.

8. GENERAL PROVISIONS

If you are located in the European Union and are using the Home-Edition Software in the European Union and not in the United States, then this XMLSPYHOME-EULA will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Home-Edition Software resides in the Handelsgericht, Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht, Wien (Commercial Court, Vienna) in connection with any such dispute or claim.

If you are located in the United States or are using the Home-Edition Software in the United States then this XMLSPYHOME-EULA will be governed by and construed in accordance with the laws of the Commonwealth of Massachusetts, USA (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Home-Edition Software resides in the federal or state courts of Massachusetts and you further agree and expressly consent to the exercise of personal jurisdiction in the federal or state courts of Massachusetts in connection with any such dispute or claim.

If you are located outside of the European Union or the United States and are not using the Home-Edition Software in the United States, then this XMLSPYHOME-EULA will be governed by and construed in accordance with the laws of the Republic of Austria (excluding its conflict of laws principles and the U.N. Convention on Contracts for the International Sale of Goods) and you expressly agree that exclusive jurisdiction for any claim or dispute with Altova or relating in any way to your use of the Home-Edition Software resides in the Handelsgericht, Wien (Commercial Court, Vienna) and you further agree and expressly consent to the exercise of personal jurisdiction in the Handelsgericht Wien (Commercial Court, Vienna) in connection with any such dispute or claim.

This XMLSPYHOME-EULA will not be governed by the conflict of law rules of any jurisdiction or the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded. This XMLSPYHOME-EULA contains the entire agreement and understanding of the parties with respect to the subject matter hereof, and supersedes all prior written and oral understandings of the parties with respect to the subject matter hereof. Any notice or other communication given under this XMLSPYHOME-EULA shall be in writing and shall have been properly given by either of us to the other if sent by certified or registered mail, return receipt requested, or by overnight courier to the address shown on Altova's Web site for Altova and the address shown in Altova's records for you, or such other address as the parties may designate by notice given in the manner set forth above. This XMLSPYHOME-EULA will bind and inure to the benefit of the parties and our respective heirs, personal and legal representatives, affiliates, successors and permitted assigns. The failure of either of us at any time to require performance of any provision hereof shall in no manner affect such party's right at a later time to enforce the same or any other term of this XMLSPYHOME-EULA. This XMLSPYHOME-EULA may be amended only by a document in writing signed by both of us. In the event of a breach or threatened breach of this XMLSPYHOME-EULA by either party, the other shall have all applicable equitable as well as legal remedies. Each party is duly authorized and empowered to enter into and perform this XMLSPYHOME-EULA. If, for any reason, any provision of this XMLSPYHOME-EULA is held invalid or otherwise unenforceable, such invalidity or unenforceability shall not affect the remainder of this XMLSPYHOME-EULA, and this XMLSPYHOME-EULA shall continue in full force and effect to the fullest extent allowed by law. The parties knowingly and expressly consent to the foregoing terms and conditions.

Last updated: 2004-09-22

Index